Combining artificial intelligence and robotic system in chemical product/process design

Liwei Cao

Supervisor: Prof. A. A. Lapkin

Department of Chemical Engineering and Biotechnology

University of Cambridge

This dissertation is submitted for the degree of

Doctor of Philosophy

Churchill College July 2021

I would like to dedicate this thesis to my loving parents

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. This dissertation contains fewer than 65,000 words including appendices, bibliography, footnotes, tables and equations and has fewer than 150 figures.

Liwei Cao

July 2021

Acknowledgements

The work detailed in this thesis was undertaken in the Sustainable Reaction Group at the University of Cambridge, Department of Chemical Engineering and Biotechnology. Many people have contributed to this work, directly and indirectly, and I sincerely thank them all, even if they are not named below. I would particularly like to thank:

Professor Alexei Lapkin for giving me guidance and support throughout my PhD and supporting me to both undertake my research and have time to develop my skills more broadly. His erudite and sharp mind is an academic role model for me, I have certainly learnt a lot and will always be impressed with his energy and ideas.

Dr Danilo Russo for his support for me both professionally and personally since joining the group and working on multiple projects with me. His passionate attitude and great cooking skills lighted me up during the down times. The drinking time every Friday night when we first started the project is definitely one of the highlight moments in my first year of PhD.

Pascal Neumann, for joining the team for his master porject and contributing to both software part and hardware part on the automated experimental system Centripeta 1.0. Kobi Felton, for his robotic and software expertise and for always being a cheerful presence to work with. Artur Schweidtmann for the discussion on the TESMO algorithm and constructive advise on the optimisation algorithm development. Jana Weber for the virtual coffee break during our write-up stage, although it is not as ideal as we planned when we first started our

PhD, which is to write up together on a stunning island. Dr Yehia Amar, for his detailed lab record and the large experimental data set for me to test and play the ML models. Zhimian Hao, Chonghuan Zhang and Ahmad Khan for the discussion on process optimisation and reinforcement learning applications. All members of the Lapkin group and Singapore CARES office, past a present, deserve thanks as they will have contributed to this PhD somehow – whether this be an idea, some technical help or just being friendly and supportive.

Outside my research group, I would like to thank all my collaborators:

Prof Vassilios Vassiliadis for the idea and constructive discussions on the development of the MINLP formulation of the symbolic regression algorithm. BASF colleagues, Dr Huanhuan Zhang, Dr Johannes Barth, Dr Werner Mauer, Dr David Lee, Zhexiong Yang and collegues from BASF both Shanghai and Ludwigshafen site, for their technical support, fresh ideas from industrial point of views on formulated product design and production. Colleagues from University of Glasgow and University of Southampton, Prof Lee Cronin, Dr Laure Points, Dr Abhishek Sharma, Daniel Salley, Graham Keenan, Prof Dave Woods, Dr Emily Matthews and her baby Owen while in her belly, for their contribution and support on the hardware and software in the EPSRC project. This project strengthened my skill from many aspects, literally.

Outside research groups, I am also grateful to have a wide support network of friends, especially my housemates, friend from the department, my college, my undergraduate studies at Tsinghua University and from back home in Beijing, China. You guys provided so much fun and some needed stress relief throughout my PhD.

Final thanks go to my parents for their unconditional love and support during my study, although you may think I was building a 'Shampoo Terminator'.

In the end, I apologise to anyone who I forgot at this time – thank you for your support, and thanks to anyone who reads this thesis.

Abstract

Product design for formulations is an active and challenging area of research. The new challenges of a fast-paced market, products of increasing complexity, and practical translation of sustainability paradigms require re-examination the existing theoretical frameworks to include the advantages from business and research digitalization. This thesis is based on the hypotheses that (i) new products with desired properties can be discovered by using a robotic platform combined with an intelligent optimization algorithm, and (ii) we can the connect data-driven optimisation with physico-chemical knowledge generation, which will result in a suitable model for translation of product discovery to production, thus impacting on the process development steps towards industrial applications. This thesis focuses on two complex physicochemical systems as case studies, namely the oil-in-water shampoo system and sunscreen products.

Firstly, I report the coupling of a machine-learning classification algorithm with the Thompson-Sampling Efficient Multi-Optimization (TSEMO) for the simultaneous optimization of continuous and discrete outputs. The methodology was successfully applied to the design of a formulated liquid product of commercial interest for which no physical models are available. Experiments were carried out in a semi-automated fashion using robotic platforms triggered by the machine-learning algorithms. The proposed closed-loop optimization framework allowed to find suitable recipes meeting the customer-defined criteria within 15 working days, outperforming human intuition in the target performance of the formulations. The framework was then extended to co-optimization of both formulation and process conditions and ingredients selection.

Secondly, I report the methods for the identification of new physical knowledge in a complex system where a prior knowledge is insufficient. The application of feature engineering methods in sun cream protection prediction was discussed. It was found that the concentration of UVA and UVB filters are key features, together with product viscosity, which match with the experts' domain knowledge in sun cream product design. It was also found that through the combination of feature engineering and machine learning, high-fidelity model could be constructed. Furthermore, a modified mixed-integer nonlinear programming (MINLP) formulation for symbolic regression method was proposed for identification of physical models from noisy experimental data. The globally optimal search was extended to identify physical models and to cope with noise in the experimental data predictor variables. The methodology was proven to be successful in identifying the correct physical models describing the relationship between shear stress and shear rate for both Newtonian and non-Newtonian fluids, and simple kinetic laws of chemical reactions.

The work of this thesis shows that machine learning methods, together with automated experimental system, can speed-up the R&D process of formulated product design as well as gain new physical knowledge of the complex systems.

Publications

The following articles were published in peer reviewed journals as the result of the work detailed in this thesis:

1. **Cao, L.**, Russo, D., & Lapkin, A. A. Automated robotic platforms in design and development of formulations. AIChE Journal, e17248.

2. **Cao, L.**, Russo, D., Felton, K., Salley, D., Sharma, A., Keenan, G., ... & Lapkin, A. A. (2021). Optimization of Formulations Using Robotic Experiments Driven by Machine Learning DoE. Cell Reports Physical Science, 2(1), 100295.

3. **Cao, L.**, Russo, D., Mauer, W., Gao, H. H., & Lapkin, A. A. (2020). Machine learning-aided process design for formulated products. In Computer Aided Chemical Engineering (Vol. 48, pp. 1789-1794). Elsevier.

4. Neumann, P., **Cao, L.**, Russo, D., Vassiliadis, V. S., & Lapkin, A. A. (2020). A new formulation for symbolic regression to identify physico-chemical laws from experimental data. Chemical Engineering Journal, 387, 123412.

The following articles were pending to submit as the result of the work detailed in this thesis:

1. **Cao, L.**, Russo, D., Matthews, E., Woods, D., & Lapkin, A. A. Computer-aided Design of Formulated Products: a Bridge Design of Experiments for Ingredients Selection

The following articles were published in peer reviewed journals as a result of work undertaken over the course of this PhD program but not detailed in this thesis:

1. Zhang, C., Amar, Y., **Cao, L.**, & Lapkin, A. A. (2020). Solvent Selection for Mitsunobu Reaction Driven by an Active Learning Surrogate Model. Organic Process Research & Development.

2. **Cao, L.**, Kabeshov, M., Ley, S., & Lapkin, A. A. (2020). In silico rationalisation of selectivity and reactivity in Pd-catalysed CH activation reactions. Beilstein J. Org. Chem, 2020, 16, 1465-1475

3. Amar, Y., Schweidtmann, A. M., Deutsch, P., **Cao, L.**, & Lapkin, A. (2019). Machine learning and molecular descriptors enable rational solvent selection in asymmetric catalysis. Chemical science, 10(27), 6697-6706.

4. Zhang, C., **Cao, L.**, & Romagnoli, A. (2018). On the feature engineering of building energy data mining. Sustainable cities and society, 39, 508-518.

Conferences

The following conference presentations were over the course of this PhD program:

1. "Towards greener production of formulation using robotic experiments driven by machine learning DoE"
   Cambridge Zero Research Symposium: Resources & Production, March 2021

2. "Optimisation of formulations using robotic experiments driven by machine learning DoE"
   AICHE Annual Meeting, Session Advancing Digitization & Sustainability in Process Development, San Francisco, CA, USA, November 2020.

3. “Symbolic Regression for the Automated Physical Model Identification in Reaction Engineering”
   AICHE Annual Meeting, Session Applications of Data Science in Catalysis and Reaction Engineering II, Orlando, FL, USA, November 2019.

4. ”Rational Solvent Selection Guided By Machine Learning and Molecular Descriptors in Asymmetric Catalytic Reactions”
   AICHE Annual Meeting, Topical Conference: Next-Gen Manufacturing, FL, USA, November 2019.

5. ”Effect of Sulfuric Acid and Solvent Environment in the Hydrogenolysis of Glycerol on ReOx-Ir Cata- lyst: A Combined Experimental and Computational Study”

AICHE Annual Meeting, Catalysis and Reaction Engineering Division, Orlando, FL, USA, November 2019.

6. ”Combining artificial intelligence and automated experiments in formulated product design”

   BASF NAO Family Day, Shanghai, China, October, 2018

7. “Classification in consumer products design”

   Statistics and Machine Learning in (Bio) Chemical Engineering, Cambridge, UK, June, 2018

Table of contents

List of figures

List of tables

List of schemes

Nomenclature

Acronyms / Abbreviations

AIC	Akaike Information Criterion
ALAMO	Automatic Learning of Algebraic Models for Optimization
BB	Branch and Bound
BIC	Bayesian Information Criterion
DAG	Directed Acyclic Graph
DDPM	Data-Driven Physical Model
DLS	Dynamic Light Scattering
DoE	Design of Experiments
DT	Decision Tress
EDA	Exploratory Data Analysis
EGO	Efficient Global Optimisation
EI	Expected Improvement
GP	Gaussian Process

GUI	Graphical User Interface
ID	Inner Diameter
KS	Kolmogorov-Smirnov
LB	Lower Bound
LHD	Latin Hypercube Sampling
MCS	Monte Carlo Sampling
MILP	Mixed Integer Linear Programming
MINLP	Mixed Integer Nonlinear Programming
MOAL	Multi-object Active Learner
MSE	Mean Squared Error
MWP	Modular Wheel Platform
NL	Number of Layers
NP	Number of Points
NTU	Nephelometric Turbidity Unit
NX	Number of Variables
PA	Protection grade of UVA
PCA	Principal Component Analysis
PEEGO	Point Exchange Efficient Global Optimisation
PNPA	Para Nitrophenyl Acetate

PNP	p-Nitrophenol
QMCS	Quasi-Monte Carlo Sampling
RF	Random Forest
RTD	Resistance Temperature Detector
SC	Symmetry breaking Cuts
SFC	Space Filling Criteria
SMCS	Stratified Monte Carlo Sampling
SPF	Sun Protection Factor
SR	Symbolic Regression
SSE	Sum of Squared Errors
SVM	Supporting Vector Machine
SV	Support Vector
TS-EMO	Thompson Sampling Efficient Multi-objective Optimization
UB	Upper Bound

Chapter 1

Introduction

1.1 Background

Formulated products consist of a blend of ingredients, processed to achieve a set of desired performance and appearance characteristics [1]. The aim of formulated product design is to find a product that exhibits a behaviour, corresponding to desired, customer-defined functional properties [1]. Formulations are ubiquitous in daily life, ranging from medicines to cosmetic creams and gels, from detergent powders and liquids to processed foods, paints, adhesives, lubricants, pesticide granules, and many more. Because of the significance of formulations markets, the developments in formulation technologies is attracting attention of both academia and industry [2]. In 2009, chemical product engineering has been introduced as the third paradigm within the field of chemical engineering [3]. The design of formulated products involves identification of target product attributes, determination of product form, selection of ingredients, development of processing steps, as well as economic and environmental analyses [4]. As a result of this conceptual and empirical complexity, research has focused on identification of a theoretical framework for formulated products design, taking into consideration all of these interlined areas [5]. Within this general framework, it is clear that access to: (i) a large number of reliable and repeatable data, and (ii) better models, would be key elements for faster and efficient formulated products development. The former challenge can be addressed by adopting robotic automated high-throughput experimentation, whereas the latter can be met by the adoption of data efficient statistical machine learning (ML) models. The automation of chemical experiments and advances in machine learning algorithms to guide automated experiments has recently emerged as a new paradigm for chemical R&D [6, 7] in robotic experimental platform for nanomaterial discovery [8, 9], design of experiments for high-dimensional statistical learning [10], synthesis planning [11, 12], discovery of reactions [13], and optimisation of process conditions through machine learning [14]. Automation and digitalisation of R&D are also offering significant advantages to discovery and optimal design of formulated products [15, 16]. The anticipated benefits stem from avoiding human bias and automating routine operations, while exploring highly-complex multidimensional input space. As a result, these approaches are particularly suited to address the new challenges of a fast-paced market, especially with the emerging constraints of sustainability and ethics, for which rapid discovery and development are fundamental requirements.

This project is initiated in collaboration with BASF SE. It aimed at developing a novel methodology, which rapidly explore chemical and material space in order to find optimal design with desired functions. This new approach have wide application from consumer products, to pharmaceuticals.

The key proposed innovation in this project is that:

1. A methodology was proposed which allows to optimise new products with desired properties using a novel robotic platform to sample the large experimental space, and by directing the random discovery process much faster by means of data-driven statistical machine-learning optimisation algorithms.

2. Physical knowledge can be generated and connected to data-driven optimization, which will result in a model, suitable for translation of product discovery to production, optimisation and control, the necessary steps towards industrial applications.

The chapters presented in this thesis address different aspects of the novelty described above and structured as follows:

In Chapter 1, existing approaches to formulated products design was reviewed, as well as the state art of automated closed-loop systems for discovery and optimization. It also states the role of the automation of chemical experiments and advances in machine learning algorithms to guide automated experiments approach within the already existing theoretical framework for formulated product design, and to discuss the different aspects of the hardware and the software to accomplish the full automation and digitalisation in the field.

In Chapter 2, a closed-loop systems was developed for the efficient optimization of the recipe of a complex formulated product of industrial interest, within which time saving and highly reproducible robotic experiments were coupled with machine learning algorithms. In the machine learning algorithm pipeline, The naïve Bayes classifier combined with the TS-EMO algorithm allowed to take into account in the optimization procedure binary discrete outputs, also avoiding waste of time and material resources. The optimization procedure outperformed human experts' intuition and suggested more convenient and low-priced solutions within 15 working days.

In Chapter 3, the optimization scheme proposed extends its application to both product and process design. The integration of product and process design is crucial due to the fact that specific formulated product microstructural attributes strongly depend on the selected manufacturing technologies and respective operating conditions. Also, the economic factor was also considered in the proposed scheme. Moreover, a bridge design approach was developed, in order to handle such problems in formulated product design, where a large a large number of components are possible to choose from, but only a few components in final composition is sought. Both of the extensions were illustrated with the shampoo case study as introduced in Chapter 2.

In Chapter 4, I report a methodology using machine learning to capture chemical intuition that researchers normally develop in their search experimentally. Sunscreen formulated products were used as the case study. Multiple feature engineering methods were applied for feature importance analysis. It is shown that with the prior knowledge learning from the system, the model prediction accuracy can be significantly improved.

In Chapter 5, a modification to the mixed-integer nonlinear programming (MINLP) formulation for symbolic regression was proposed with the aim of identification of explicit physical models from noisy experimental data. The methodology was proven to be successful in identifying the correct physical models describing the relationship between shear stress and shear rate for both Newtonian and non-Newtonian fluids, and simple kinetic laws of chemical reactions.

In Chapter 6, an overall conclusion as well as the outlook on future work is provided.

1.2 Theoretical Framework for Formulated Products Design

During the past decades, enormous efforts have been made in order to develop methods and tools for product design and development in various disciplines, such as material science, chemical engineering, marketing and management. Ng, Gani and Dam-Johansen identified the following three approaches for product design, which are classified based on their solution strategies [17]:

1. Experiment-based trial-and-error approach.

2. Physical model-based approach.

3. Integrated experiment-modelling approach.

These three approaches are discussed in the following. The experiment-based trial-and-error approach is the preferred and most common one for the design of formulated products. By performing experiments at all steps during the development of a formulation, the product with desired properties can be developed. One previously reported explicative example is the development of the inkjet formulation [18]. In this case, with only few key ingredients, and a set of typical solvents and dispersants, it was possible for an experienced researcher to develop the optimal blend on a lab-scale, and eventually use the gathered experimental data to generate a model for future use [19]. However, this approach suffers from two main drawbacks: (i) it requires a large amount of resources and is highly time-demanding, (ii) it is critically dependent on the level of expertise of an experimentalist, the past knowledge, both formal and tacit as identified by Chandrasegaran et al. [20]. In particular, tacit knowledge, consisting of subjective insights, intuition and heuristic qualitative rules is not easily transferable and is usually lost with the loss of the experts in product development. Therefore, this approach would be beneficial if the number and the type of ingredients and processes conditions were limited *a priori* and skilled experts are involved in the process. On the other hand, computational methods, i.e. physical model-based design of formulations, were proposed in order to reduce the experimental cost and to speed up the R&D process. In the last few years, various attempts have been made to establish systematic methodologies. Computer-aided methods have been proposed for solvent design [21], mixture design [22], general molecular design [23] and etc.. A review of computer-aided molecular design (CAMD) methods for product and process design was published by Gani [4], while Ng et al. reviewed significant developments, current challenges, and future opportunities in the field of chemical product design using the CAMD tools [24]. A key concept in CAMD is to utilize different chemical property models for possible chemical species in the pool, formulated as a mixed-integer linear/nonlinear programming (MILP/MINLP) optimization model, then solved with numerical optimization techniques [2]. The suitability of these structures for a particular task or process can be evaluated with respect to a chosen criterion (for instance, the solubility of the target compound), while considering physical and chemical constraints, as well as process constraints of varying complexity. From the solution of the optimization model, the optimal product is obtained. It has been first applied to the design of single molecule species, and witnessed a huge success. The applications vary from the design of refrigerants [25] to surfactants [26]. The CAMD method was then extended to the design of mixtures and composite chemical products, and identified as computer-aided blend design (CAM^{b}D) [27] also reported as computer-aided mixture design (CAM^{x}D) [28, 29]. Typically, almost all CAMD/CAM^{b}D methods use group contribution (GC) based property prediction methods [30, 31] to evaluate the generated compound with respect to the specified set of desirable target properties. UNIFAC [32, 33] and SAFT-γ [34] demonstrated to be accurate and useful in calculating solubility, phase equilibrium, partition coefficients, and various other properties. However, one significant issue is that they rely on binary interaction parameters for every pair of groups in solution, often not available in thermodynamic properties databases [35]. An alternative way is using quantum chemistry calculation for thermodynamics estimation. The COSMO-RS and COSMO-SAC are two of the most popular post-processing methods in COSMO solvation model, where the estimation of thermodynamics only relies on the composition-independent charge density distributions, also known as sigma profile, and molecular volume. Detailed review on those methods can be found in Refs. [29, 36, 37]. Furthermore, a systematic review on available computer-aided methods and associated software tools for formulated product design can be found in Ref [38]. Briefly, the model-based approaches are able to efficiently find feasible candidates within the application range of the available models. However, since the function-materials-structure-processing relations have not been developed for complex formulations, including the ones determined by nano or microscale, some target properties are hard to predict with computational tools only [39].

As a result, the third and final integrated experiment-modelling approach was proposed, which consists in combining the computer-aided model-based techniques with heuristic-based experimental testing and improvements of the formulation design. The integrated approach usually consists of three stages: the problem definition stage, the model-based design stage, and the experiment-based verification stage [40]. In the problem definition stage, the targets are translated into a set of thermophysical properties and into a list of categories of ingredients, which are to be included in the formulation via a robust knowledge base. In the model-based design stage, structured databases, dedicated algorithms, and property physical model library are employed for designing a candidate base case formulation. Finally, in the experiment-based verification stage, the properties and performances of the proposed formulation are tested experimentally. Through this systematic list of stage action, the formulation is then validated.

By limiting candidate formulations to be tested and verifying the design in the last stage, the integrated approach is convenient by saving the time and resources (compared to experiment-based trial-and-error approach) and increasing the accuracy of the results (compared to physical model-based approach).

In this framework for formulation design, the integration of robotic experiments and statistical ML models would be a further step in the improvement of the integrated approaches. In this sense, this approach would combine the time and resource efficiency of robotic platforms with the fact that predictions of statistical models are only based on data, with no need of extensive first principles physical knowledge.

The approaches reviewed so far are related to the core product design, and they are based on the assumption that the target properties and the final market destination have already been identified and analysed. That is, the core design approaches are part of a broader theoretical framework, taking into account different co-existing levels within a complex decision making hierarchy. These have been proposed by several papers and first reviewed by Gani and Ng in 2015, focusing on product conceptualization [41]. Interesting hierarchical models integration in this broader framework was identified by Fung et al., who proposed a grand model for chemical product design, which indicates the relationships between different aspects [1]. It consists of a process model, a property model, a quality model, a cost model, a pricing model, an economic model, as well as factors such as company strategy, government policies, and regulations. Further elements have been added by Seider et al., proposing a model considering issues such as sustainability, company strategy, aesthetics, human senses, and so on [42], while Zhang et al. included supply chain analysis for optimization of selection of product ingredients [43]. Despite the large number of problem aspects included in the grand structure, there are five common key elements included in these high level integration approaches [5]. For the modelling part, a physico-chemal model (material properties, product structure, process condition and etc.), a rule-based model (supply chain, economics analysis) and databases are involved. For the experimental part, experimental and analytical tools are also considered within the superstructure for the validation of design.

In Fig. 1.1, we illustrate the integration of the methodologies described in this thesis in the pre-existing theoretical framework reported by Zhang et al. [38]. Briefly, the market needs to define the product and its desired properties, that can be translated into quantifiable properties functions. Once identified, the next step in the general product design is to analyse the existing knowledge in terms of preliminary information, tacit knowledge in the form of operators' expertise, and formal knowledge, derived from first principles and the available models, to define the objective functions to optimize. It is important to stress that commercial formulated products are often complex mixtures for which no predictive models are available, and the complex interactions between different ingredients and process variables are not easily translatable into predictions of the final properties, even by experienced formulators.

The most common situation would be the availability of a small preliminary data set, which can be used to define reasonable constraints in the input variables space. The preliminary data can be used in combination with DoE techniques to run a first batch of experiments to maximize the information gain and start the iterative process for the multi-optimization problem. In this regard, there is still an urgent need to have fast DoE algorithms to maximize the information on different conflicting continuous and discrete targets at the same time. The lack of predictive models for most of the complex interactions and properties of formulated products suggests the superiority of DoE algorithms based on the use of data-driven surrogate statistical models, as discussed in Section 1.3.3. Based on predictions of such models, trained on the available experimental data, robotic platforms can quickly and reliably run experiments to generate samples. Automated analytics can then generate a new data set and use it to iteratively train the algorithms until satisfactory conditions are found. The underpinning assumption is that the adoption of automated machinery and statistical models would significantly speed up the discovery of new products and the optimization of the conditions leading to a faster release of the product in the market.

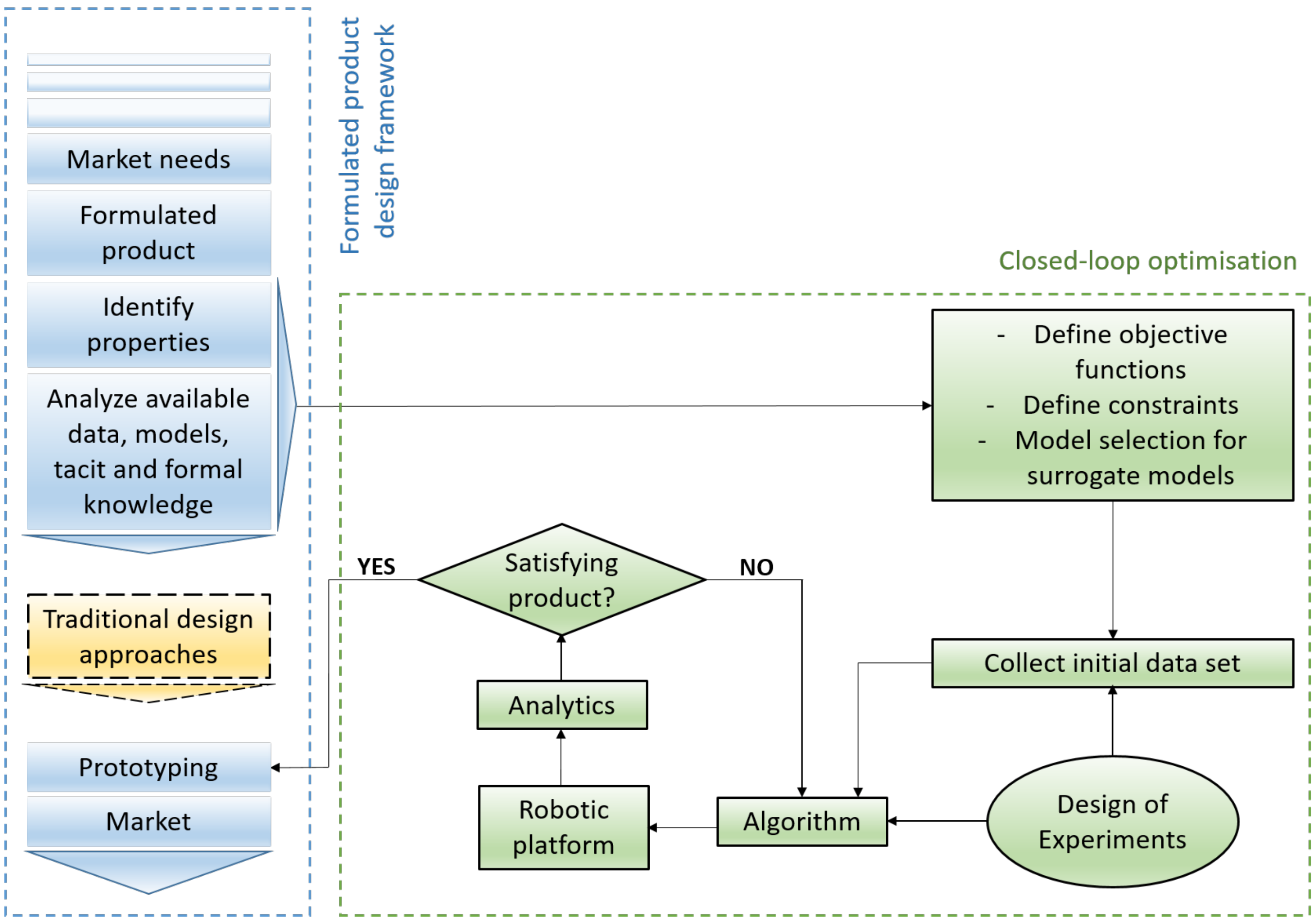


Fig. 1.1 Integration of closed-loop robotic platform in the general framework for formulated products design. (Figure adapted from Zhang et al. [44])

1.3 Closed-loop systems for formulations: state of art and challenges

The recently reported combinations of process automation, artificial intelligence and statistical models in closed-loop optimization are potentially very promising in reducing the release time of new products in the market without the need for a detailed physical knowledge of new complex systems [45, 46, 10]. However, at present, only a very few papers started to address the challenges and applications of computer-guided closed loop optimization in the field of formulated products and there is a general lack of discussion on the role of these new tools in the more general framework of product design, already outlined in the literature. In this Section, a brief overview of the existing techniques were provided, identifying the challenges for future research. For a thorough review of closed loop optimization in the fields of chemistry and chemical engineering the reader can refer to Mateos et al. [46], Houben and Lapkin [47] and Horbaczewskyi et al. [48].

The common features of automated closed loops reported in the literature are Fig. 1.1:

1. A robotic platform to run experiments in an automated fashion.
2. An automated on-line and in-line analytical tools to evaluate the outcome of experiments.
3. An algorithm suggesting new experiments to carry out, based on the predictive results of surrogate models, ideally cheap to evaluate. The trade-off between exploitation and exploration is of crucial importance in reducing the time and the resources needed for product development.

1.3.1 Robotic Platforms

First automated hardware for chemistry can be dated back to the late 1960s [49]. Since then, considerable advances have been made to expand the potentialities of such a tool. For a detailed historical excursus, the reader can refer to Ref [49]. Robotic platforms proposed in the academic literature are mostly developed in the field of chemical reaction and very little has been proposed for the generation of complex formulated products, for which physico-chemical interactions between the ingredients have less obvious outcomes compared to reactions between chemical species. The state-of-art hardware can be grouped into two main categories: (i) automated continuous microfluidic platforms and (ii) modular batch operations [50]. Advantages and drawbacks of both configurations are widely discussed in the existing literature [51, 52] and are beyond the scope of the present paper. However, it must be highlighted that, whilst the former continuous flow devices seem extremely promising for investigating reaction conditions in an efficient and resource-undemanding way, the latter possess great advantages for the mixing, the processing of emulsions, handling of solids, and for the investigation of thermodynamics-related properties [53], e.g. stability. For most formulated products, the process determining the final structure, thermodynamic state, and properties, consists in a combination of rigorously controlled mixing of ingredients at a certain temperature, and stepwise addition of different ingredients at different stages of the process. Therefore, the main challenges for the application of automated hardware to formulated products design, can be identified as:

1. producing a large number of batch samples in a relatively short time;
2. the stringent control of mixing and temperature;
3. accurate handling of solid ingredients;
4. transferring samples between different bays for different unit operations (dispensing, processing, analysis, etc.);

5. standardized flexible robotic hardware that can be easily adapted to the specific workflow.

High-throughput production of batch samples has developed at different speed and with different purposes and philosophy in industry and academia. Interesting automated systems with applications for formulated products are represented by the robotic platform currently developed by Unilever and the University of Liverpool, and the FORMAX system, proposed by ChemSpeed Technologies. The latter seems to be one of the more flexible platforms on the market for formulation preparation; it consists of 30 exchangeable robotic tools and up to 36 formulation vessels. Each formulation vessel is equipped with a stirring system with speeds up to 6,000 rpm and precise temperature control. They also include liquid and gravimetric solid handling, high viscous liquids dispensing, high shear homogenization, and other robotic features like capping, crimping, gripping. Another commercially proposed solution is the GEOFF automated formulation robot by LABMAN, with a productivity of 24 formulations per day. Despite great potential, the lack of academic papers describing in detail such systems or adopting them to exploit their full functionality suggests that their integration into existing laboratories is not always straightforward, affordable, or convenient. This poses a serious question about the "democratization" of such tools for their easy exploitation and wider impact in research.

In this direction, pioneering work in the development of batch modular systems has been described by Cronin and co-workers [54, 55, 52] that could be adapted to the specific requirements of this type of products. To date, the developed hardware has only been used to investigate chemical reactions, with the only exception of the study of physical interactions determined by thermodynamics, which then manifests itself in complex dynamic behaviour of oil droplets in a continuous water phase [56, 53]. Being developed for different purposes, these platforms are only able to produce one sample at a time with interstage automated cleaning of the reactionware/containers. An attempt to overcome this limitation can be found in the recent studies [57], where an automated rotating wheel, coupled with a 3D-printed dispensing element and automated syringe pumps, can allocate batches of 24 vials per run. The potential of using 3D printing technologies to build inexpensive hardware was also highlighted [58].

Systems for carrying out reactions in parallel under different conditions have been implemented by Chemspeed, Hel Ltd., and other vendors [59–76]. Specifically, high- and medium-throughput facilities have been implemented in the pharmaceutical industry, where large numbers of compounds and reactions should be screened. Some examples are represented by high-capacity storage facilities handled by moving robots and/or robotic arms and miniaturised prototypes for synthesis and testing [77]. Researchers at Merck [78] have recently proposed a 96-well metal microtiter plates to screen large numbers of reactions on a small scale, still highlighting that automated liquid handling requires significant investment and training, whereas solid handling is both slow and inaccurate. Novartis Pharma developed a high-throughput robotic system based on the use of deep well microplates with up to 480 samples [79].

Dispensing of ingredients is followed by or is simultaneous with processing of the mixtures. In most academic papers, mixing seem to be efficiently automated using magnetic stirrers activated by software-controlled magnets [58, 52]. However, in all the presented solutions, temperature control is not ensured and, in some cases, mixing appears to be far from ideal, stressing the need for standard thermostated mixing devices to explore different effects of mixing and shear stress on the final product. In this sense, continuous flow microdevices might be advantageous due to the fine control of the shear stress and the flexibility in attaining different mixing regimes on a small-scale. However, also in this case, the challenge of reaching similarly high mechanical stresses typical of a shaking process is still an issue, also considering that most formulated products are processed on a long time scale not easily achievable in continuous flow micro-devices. Some continuous flow high-shear mixers have been proposed [80–82], also for formulated products.

To the best of our knowledge, the only reported example in the literature of closed-loop robotic optimization for formulated products [83] involves the use of an off-line non-automated incubator to efficiently process samples, once again demonstrating that this remains an open challenge in academia. Integration of all the main aspects of sample preparation, i.e. liquid and solid dispensing, mixing, heating, and weighing, in the same platform has been achieved in some commercial systems, the most notable platforms being Chemspeed and Symyx [84], GEOFF automated formulation robot by LABMAN.

Assuming that different operations cannot be easily carried out in the same part of an automated platform, the next problem to address is the automated transfer of samples from a bay/station to another. Li et al. proposed the use of automated guided vehicles for transferring operations in automated labs [85]. Another common solution, robotic arms with multiple degrees of freedom, is not without its own challenges, since these are often expensive and designed to carry out only specific repetitive operations with a reduced flexibility. The problem was effectively solved in the work by Steiner et al. [52], for reactive mixtures: reactive mixtures were transferred from a batch unit operation to another (reaction, separation, purification and etc.) by pumping through automatically controlled connection channels. In the field of synthetic chemistry another industrial example of automated laboratory can be found in the system developed by Aventis Pharma [86], based on the use of robotic transferring shuttles on a rail system between different work stations. The automated lab could efficiently carry out synthesis, mixing and temperature control, and auxiliary operations such as weighing, capping and uncapping, as well as separation operations, like liquid-liquid extraction, evaporation, filtration and drying. However, formulated products pose new challenges, such as the flow of highly viscous products, the presence of dispersed solids, multiple heterogeneous phases, and the formation of inter-molecular structures (core-shell particles, entangled molecular chains, micelles, etc.) that might be dramatically affected by the shear effects in continuous flow.

Solids handling is of primary importance in automated platforms for the optimization of formulated products, considering that solid ingredients are often dissolved in a liquid matrix or present in the final product in the form of solid dispersions or solid blends [87, 88]. However, none of the reviewed papers in the field of robotic platforms for the optimization, discovery, and development of chemical processes has efficiently addressed the issue, despite the fact that it would be beneficial also for chemical reactions. Several technical solutions commercially available in drug discovery had been described [89]. Trap-door mechanisms with holes of different diameter are described as surprisingly accurate methods of solid dispensing, although inflexible and dependent on the packing of the material in the holes, which makes it unreliable for powders of different size distributions. Solid handling pipettes rely on vacuum [90], or electrostatic forces [91] which makes them difficult to install and limits their applicability, whereas more traditional devices are based on the use of Archimedes screws [87]. Other commercially available solutions are included in the Chemspeed's gravimetric dispensing unit, the Mettler-Toledo's dispensing stations, and the Zinsser Analytic Calli robotic powder handling; at present, there are no publications reporting their integration in robotic platforms for reaction and formulations development and is was recently stressed that no general automated solid handling solution currently exists [78].

One final remark is the need for standardized unit platforms to be combined together in different ways for a faster and cheaper exploitation of robotic laboratory technology. Standardization is a fundamental requirement for commercialisation and application of technologies on a large scale. Moreover, standardization of robotic hardware and software would enable faster implementation of communication between different platforms, efficient collaboration between researchers with different backgrounds, and creations of networks of different platforms working on the same task, even remotely. The need of modularity [51, 52] and standardization [54] has been already highlighted and partially tackled in the field of reaction development. However, future research is fundamental to extend the range of applications and to make this relatively new tool accessible, available, and usable for scholars with different expertise.

1.3.2 Analytics

Automated analytical tools are of crucial importance for the fast and efficient adoption of robotic platforms for formulation development and they represent the ongoing bottleneck to the wide adoption of such systems in academia and industry. Once again, most of the analytics automated in the literature has been used for reaction development. A thorough review can be found in Mateos et al. [46] and Houben et al. [47]. The most common adopted techniques for reactive systems are UPLC and HPLC [92, 93, 51, 94–98, 45, 99–103, 10, 104, 105], GC [106, 99], MS [37, 40, 46, 56, 52, 54, 53, 105, 107], IR [51, 55, 107, 108, 99], Raman [51], and UV spectrophotometry [106, 109, 110]. However, according to the theoretical framework outlined by Bernardo et al. [39], formulation design is a cycle of inversion and evaluation of quality, property and process functions. Unlike reaction optimization, in this case, most of the quality and property functions cannot be easily parametrized and most of the measurable final properties of products are not easily correlated to the concentration of chemical species in the system, meaning that there is a need for more complex automated characterization of the obtained samples.

For formulated liquid products, the main general desired properties can be identified as: stability, aspect (colour and turbidity), viscosity, surface tension, pH, conductivity, zeta potential and droplets size distribution, in the case of emulsions. Therefore, more complex, analytical sensors need to be identified and integrated in the robotic platforms to acquire data about different properties at the same time. Other important properties can be functional performances. i.e. for example UV protection of solar creams, or other sensory properties like odour, stickiness, etc. A first step in the automation of sensory properties measurement is represented by the new robotic tactile systems SynTouch (https://syntouchinc.com/technology/).

One key property of several commercial formulated products, ranging from detergents to personal care products, is their external appearance, which can be quantified using discrete and continuous variables. The former can be defined as "stability" which is related to the capability of the system to not show phase separation, whereas the latter can be quantified considering their absorbance spectra in the visible range and their turbidity value, measured in Nephelometric Turbidity Unit (NTU). Phase separation can be evaluated through automated image processing from automated cameras. Automated cameras and image processing coupled to robotic platforms have already been proposed in other contexts [56, 53].

Very recently, Cao et el proposed a robotic platform to measure turbidity of a commercial detergent, based on the adoption of a cheap LED and a light sensor on a moving 3D-printed support [83]. The LED and the light sensor are fixed on the opposite sides of a vial containing the samples, on a rotating wheel accommodating up to 24 vials. The electrical signal can be converted to the turbidity value in NTU, based on a calibration with turbidity standards. Following an analogous protocol, moving probes have been proposed to automatically measure pH values [57] and the same can be applied to conductivity measurements. pH measurements can be also carried out by the FORMEX platform by Chemspeed Technologies and the automated GEOFF Labman platform. At present, there are a few examples of closed-loop optimisation optimizing particle size [14] and viscosity [83, 111], whereas no example of zeta potential and surface tension in such loops have been found to date. However, in the few above-mentioned examples, both dynamic light scattering (DLS) and viscosity measurements were manually carried out offline. A promising alternative to carry out DLS and Zeta potential analysis in an automated fashion seems to be represented by the new Malvern Zetasizer coupled with an automated autosampler, however, there are still no examples of integration of such pieces of equipment in the automated platforms at present. Automation of DLS analysis was reported by Zhao et al. [79], using a DynaPro light scattering system (Wyatt Santa Barbara, California, USA), with an average throughput of 30 samples per hour.

Automated viscosity measurements can be a challenging task, especially for high viscosity and non-Newtonian fluids. One example of a semi-automated capillary viscometer can be found in Neumann et al. [110]. In this case, viscosity is calculated from the measurement of pressure drop in a capillary in which the fluid is flowing. An automated syringe pump can dispense the fluid sample and the system can perform cleaning cycles with a solvent in between the measurement runs. The described device has been successfully adopted to characterize rheology of both Newtonian and non-Newtonian low-viscosity fluids. Desmukh et al. [112] also proposed a similar system, based on the analysis of the mass flow behaviour or modelling of the pressure profile along the tips of multiple pipettes. In this case, it is claimed that the system is rapid and parallelized, allowing analysis of more than 100 samples in less than an hour, although accurate testing was only shown for Newtonian fluids. The main challenges associated with this type of devices are the narrow pressure range and solvent compatibility of pressure sensors, the accurate temperature control, and the need for smooth and pulseless dispensing to have accurate measurements. Further research will have to extend the range of usability of these devices and to integrate them in more comprehensive work flows, for the autonomous production and analysis of liquid formulations. An alternative solution is represented by the coupling of robotic arms with traditional rotational viscometers [113]. Commercially available systems for automating viscosity measurements are illustrated by the GEOFF robot and the Phil CUP/BOB rheometer by LABMAN automation, and the high-throughput rheometer Anton Parr HTR 502.

Finally, it is worth mentioning other examples of high-throughput automated assessment of less obvious and easily quantifiable properties of formulated products reported in the literature: among these: the dirt removal efficacy of different cleaning systems [114], and colour, glossiness, homogeneity, friction, and other mechanical properties of coatings proposed by the FORMAX platform by Chemspeed Technologies (https://www.chemspeed.com/formax/).

1.3.3 Algorithms

Robotic platforms can iteratively provide data points to train DoE algorithms, suggesting new conditions in order to optimize the input variables with respect to one or more target functions.

There are mainly two large groups of DoE algorithms: static and adaptive [115]. The static sampling techniques, also known as one-shot sampling, is a type of method wherein all the sample points are generated at once [116, 117]. Depending on the understanding of the system and the computational power, it can be further classified into system-free design and system-aided design. The key criterium for the system-free DoE is its space-filling ability. Factorial design, fractional factorial design are the classic system-free DoE methods, which aim to fill the space uniformly. To add randomness in the filling procedure, Monte Carlo sampling (MCS) was proposed, which uses pseudo-random numbers to generate sample points for space-filling. It is then further developed into stratified Monte Carlo sampling (SMCS), Quasi-Monte Carlo sampling (QMCS) and so on, in order to overcome shortcomings of the classic MCS method. Abundant literature from the fields of mathematics, statistics and engineering exists for Monte Carlo type of sampling techniques [118–121]. These methods inherently aim to space-fill but lack quantification of space-filling during the placement, such as uniformity-based space filling criteria (SFC) [122, 123], and distance-based SFC [124, 125]. Therefore, various methods which use the SFC as the objective function in the placement optimization problem were proposed. McKay et al. developed Latin Hypercube Design (LHD) also known as Latin Hypercube Sampling (LHS). It was inspired by the concept of Latin square sampling [126], where an n-by-n matrix is filled with n different objects such that each object occurs exactly once in each row and exactly once in each column.

Further, its variations such as orthogonal array-based LHS [127], orthogonal LHS [128], and symmetric LHS [128] were also introduced. Johnson et al. proposed two distance-based designs: maximin and minimax [125]. The maximin design maximizes the smallest distance between any two points; similarly, the minimax designs minimize the maximin distance between two points.

Although the system-free DoE techniques are easier to implement and less computational power is needed, researchers realized the vital importance of incorporating system information while generating experimental designs. To generate system specific design, scholars proposed model-based designs in different ways, such as maximum entropy sampling, mean squared error (MSE)-based designs. Lindly [129] proposed a measure to quantify information based on Shannon's entropy [130]. This entropy criterion was first employed by Shewry and Wynn to construct system-based designs [131, 132]. The MSE-based design is first employed by Sacks and Schiller [133] as the prediction accuracy of a surrogate model can be improved by minimizing its integrated mean squared error [134].

Adaptive sampling, also known as sequential sampling has attracted attention from both research and industrial community. It can overcome the under/oversampling and poor system approximations resulting from the static sampling methods [135]. It has been also shown within numerical analysis that the adaptive sampling methods yield superior surrogate approximation and lower computational expense compared to static techniques [135]. Researchers have reported adaptive sampling techniques for different surrogate model, such as support vector machines (SVM) [136, 137], artificial neural network (ANN) [138], and others [139–141]. These are the type of adaptive sampling techniques where sampling points are placed systematically, yet still stochastically. In contrast, methods which formulate optimization problem to place new samples were also reported. Cozad et al. [142] proposed ALAMO algorithm (Automatic learning of algebraic models for optimization), which is a surrogate modelling tool where a derivative-free optimization problem is solved to maximize deviation of the surrogate model prediction error in order to place the next sampling point [143, 142]. Detailed review of the existing algorithms has been published [115]. Here we outline the general needs of the formulated product development and identify the key aspects for future research.

1. Formulations can be complex physico-chemical systems for which no existing physical models are readily available. The use of cheap-to-evaluate black-box surrogate statistical models is particularly suited to model the responses of the products to variations in the input space.

2. In formulation design, it is extremely important to consider multiple, often conflicting, targets and performance criteria. There is no commercially relevant formulation which does not have to meet several targets at the same time in terms of final properties (aspect, fragrance, touch, viscosity, stability, etc), costs, and environmental impact. In this sense, several of the proposed single-objective optimization algorithms are completely inadequate. Combining several targets in one single objective, i.e. scalarizarion, is a possibility as shown for example in the multi-objective active learner (MOAL) [144] methods. However, this is not ideal, since it requires prior knowledge, introduces bias, and often is not straightforward [145]. Successful implementation of multi-target optimization has been so far achieved for continuous variables using the Thompson sampling efficient multi-objective algorithm (TS-EMO) [146, 10].

3. The sustainability challenge imposes targets for rapid development of new formulations or substitutions of some ingredients with others, as environmental legal requirements and consumers' ethics become more and more stringent [147]. As a result, algorithms need to be fast and models cheap to evaluate, also in exploring a high-dimensional combinatorial space. In addition, both discrete and continuous variables and target performances need to be efficiently taken into account at the same time.

4. The main drawback of black-box surrogate models is that they generally do not provide any information about the physics underpinning product's functional performance. In this sense, the use of data collected from closed-loop optimization procedure for generation of physical knowledge is crucial in gaining a better understanding of the processes to rapidly adapt and transfer the results to similar systems. Very preliminary results in this sense can be identified in the physical interpretation of models hyper-parameters [10], the manual interpretation of Pareto fronts by human experts [148], and, more recently, the automated capture of chemical intuition transferred between similar systems [149], and the automated generation of physical laws from data [110].

5. As in the case of hardware, there will be a general need for user-friendly open-source software interfaces, to enable experimentalists to apply the developed techniques regardless of their specific field of expertise and democratize the use of such tools.

Chapter 2

Formulation optimization using robotic experiments driven by machine learning DoE

2.1 Introduction

Formulated products are complex mixtures of ingredients, whose time to market can be difficult to speed, due to the lack of general predictable physical models for the desired properties. In this chapter we present such closed-loop optimization system as introduced in Section 1.3 for the multi-objective optimization of a commercial formulated product. Here we report the coupling of a machine-learning classification algorithm with the Thompson-Sampling Efficient Multi-Optimization (TSEMO) algorithm for the simultaneous optimization of continuous variables enables to simultaneously meet discrete (namely, formulation stability) and continuous (namely, viscosity, turbidity, and price) targets. The methodology is successfully applied to the design of a formulated liquid product of commercial interest for which no physical models are available. Experiments are carried out in a semi-automated fashion using robotic platforms triggered by the machine-learning algorithms. The procedure allows to find 9 suitable recipes meeting the customer-defined criteria within 15 working days, outperforming human intuition in the target performance of the formulations. The proposed methodology enables to find suitable solutions within a relatively short time, i.e. 15 working days, using little empirical prior knowledge about the physical system to define the constraints of the input variables. This makes the proposed pipeline particularly suitable for the early stages of the formulated products design.

2.2 Materials and Methods

2.2.1 Case study and materials

The case study under consideration is a commercial formulation consisting of three different commercially available surfactants (S1 = Texapon SB3, S2 = Dehyton AB30, and S3 = Plantacare 818), a polymer (P1 = Dehyquart CC7), a polymer (P1 = Dehyquart CC7), and a thickener (T1 = Arlyon TT). The pH was adjusted using citric acid (ACS reagent, $\geq 99.5\%$) from Sigma-Aldrich, used as received. Turbidity standards (1, 2, 5, 10, 100, 500, and 1000 NTU) were purchased from Sigma-Aldrich.

2.2.2 Close-loop optimization procedure

A general scheme for the optimization procedure is given in Fig. 2.1. In Fig. 2.1, continuous lines represent the materials flow, whereas dashed lines represent the information flow. The formulation was simultaneously optimised with respect to viscosity, turbidity, stability and price. At each iteration, a batch of 8 different suggested samples is prepared using the Robot R1. Each batches were run twice to ensure repeatability. The so prepared samples are then processed to generate the final product. The samples are successively transferred to the

Robot R2, which can automatically perform pH, turbidity and stability tests. The samples are finally analysed off-line to measure viscosity. Details on the robotic platforms R1 and R2, and the experiments are provided in Section 2.2.3. The turbidity and viscosity values are used to train surrogate Gaussian Processes (GPs) models for prediction of the target outputs. The price is analytically calculated using the unitary price of each ingredient and their relative amounts. Based on the predictions, the TS-EMO algorithm generates 8 new conditions to be tested in order to find a compromise between exploitation (finding the best conditions to minimize the objectives) and exploration (reducing the uncertainties) of the input chemical space. The generated temporary suggestions are then tested in silico using a classification algorithm to predict which samples would be stable. The conditions that give unstable formulations according to the classification algorithm are discarded, and the TS-EMO algorithm is reused to generate other suggestions, until an entire batch of 8 stable conditions is available. The new suggested conditions are finally added to the data set and used to trigger a new iteration. Details about the TS-EMO algorithm and the classification algorithm are provided in Sections 2.4 and 2.5. The file repository used for this work can be found to the following GitHub page: https://github.com/sustainable-processes/centripeta.

2.2.3 Robotic platform

Samples preparation and analyses were partially automated by using two modular wheel platforms (MWPs) adapted from adapted from Sally et al. [150] with modification to the needs of this projects work. The original platform publication describes the base model as well as certain additional optional modules. This project utilised a number of these modules to complete the workflow. Fig. 2.2 shows a picture of the adopted experimental set-up.

Briefly, both platforms consist of a laser-cut rotating wheel which can allocate up to 24 sample vials per batch. In R1, a 3D-printed element, equipped with a variable number

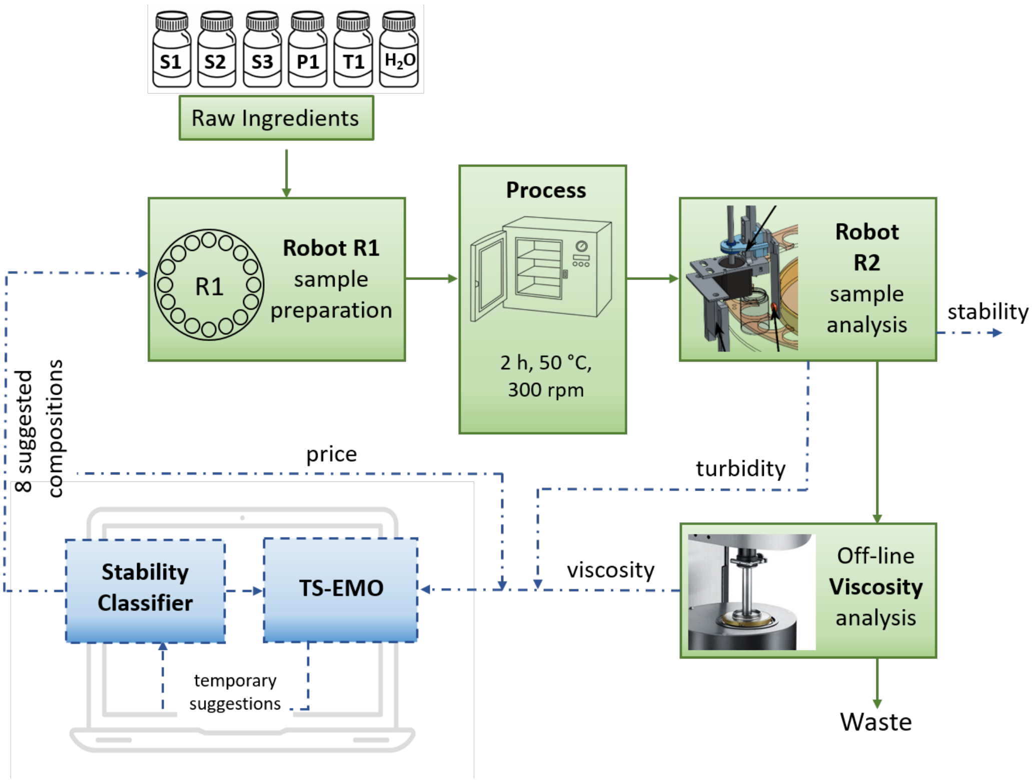


Fig. 2.1 Scheme of the adopted closed-loop optimisation workflow.

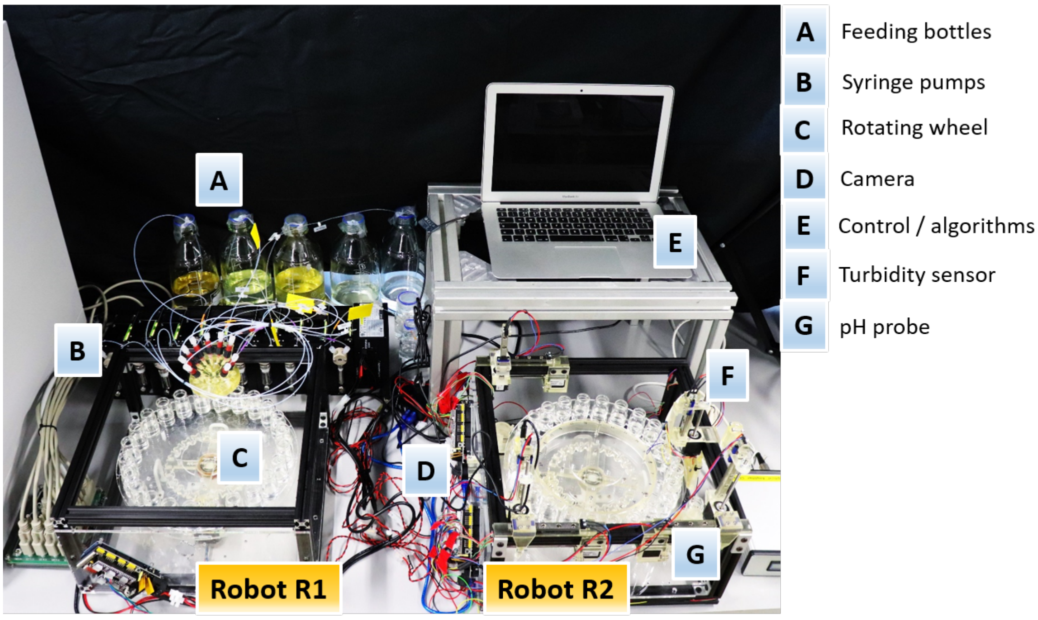


Fig. 2.2 Adopted experimental setup.

of needles, is connected to automated syringe pumps (Tricontinent, Gardner Denver, C-Series), dispensing the five different ingredients. The three surfactants are pre-diluted in water to have a concentration of active matter of 20 $g \cdot L^{-1}$ in the feeding bottles. pH was pre-adjusted to the desired value of 5.5 using citric acid. The requirement for the pH value was fixed for the specific application of the commercial product under consideration. The polymer P1 and the thickener T1 were used as received. The system was automated and triggered by a Python script to generate 8 samples at each iteration of the optimization procedure. Samples were always prepared according to the following constraints for the concentrations, defined using the little semi-empirical pre-knowledge about the system: $S1 + S2 + S3 = 15\ g\ (active\ matter) \cdot L^{-1}$, $P1 \leq 2\ g \cdot L^{-1}$, and $T1 \leq 2\ g \cdot L^{-1}$. The generated samples were then transferred to an incubator (Corning LSE 71L Shaking Incubator) where they were mixed for 2 h, at 50 $°C$ and 300 rpm. The obtained formulations were cooled down to room temperature before being transferred to R2. In this second robotic platform, we perform three kind of analyses in an automated fashion at the same time. A pH probe confirms that no pH changes occur during the process (VWR pH electrode, semi-micro, pellon junction 662-1767). No significant deviation from the target pH = 5.5 was recorded at any time. A built-in turbidity sensor is used to measure the turbidity value in NTU. Calibrations with turbidity standards were carried out every three days. Finally, an automated camera was used to take pictures of the samples and discriminate between stable homogeneous samples and unstable formulations presenting phase separation. The samples were tested off-line to measure the viscosity at a shear rate of 10 s^{-1} and 25 $°C$, by using a rotational viscometer (ARES Rheometric Scientific, strain controlled, couette configuration). The viscosity tests were run offline due to the fact that current auto-viscosmeter cannot handle the high viscosity of the shampoo formulation in this case study [151]. However,it can still demonstrate the key points in a closed-loop optimisation scheme for works on overcome the current issue in automated analytical tools.

Platform - General overview

The first platform “R1” was used for sample preparation and was an exact replica of the one reported in Sally et al. [150], the only exception being the number of possible ingredients to dispense as well as the pump type used (syringe pumps). This platform is a simple liquid handling station driven by a Geneva drive to ensure accurate placement of each reaction vial. The general instructions for assembling of a base model of this platform can be accessed in the following GitHub page containing:

- STL file for 3D printed parts;
- DXF files for laser cut components;
- Instructions for the assembly of all components;
- Software files and accompanying instructions for installation and usage;
- Bill of materials for commercially available components and custom ordered pieces.

A detailed description of this platform can be found in Ref. [152], as well as the Github link (https://github.com/croningp/InorganicClusterDiscovery). For the purposes of this work, we have described our custom set-up and how it operated. The second platform “R2”, designed for sample analysis, was based on the same wheel system, this time equipped with two X/Y/Z movement modules constructed using commercially available components and 3D-printed parts. The first of these modules was fitted with an LED light source and a light detector for turbidity measurements, the second was equipped with a pH probe. The platform was also equipped with a camera for image collection.The detailed design of the platform hardware were shown in Fig. 2.3.

Samples are dispensing directly into 14 mL vials via TriContinent C-series syringe pumps (a full wheel containing 24 such samples) on platform 1, Fig. 2.3 A. Samples are removed

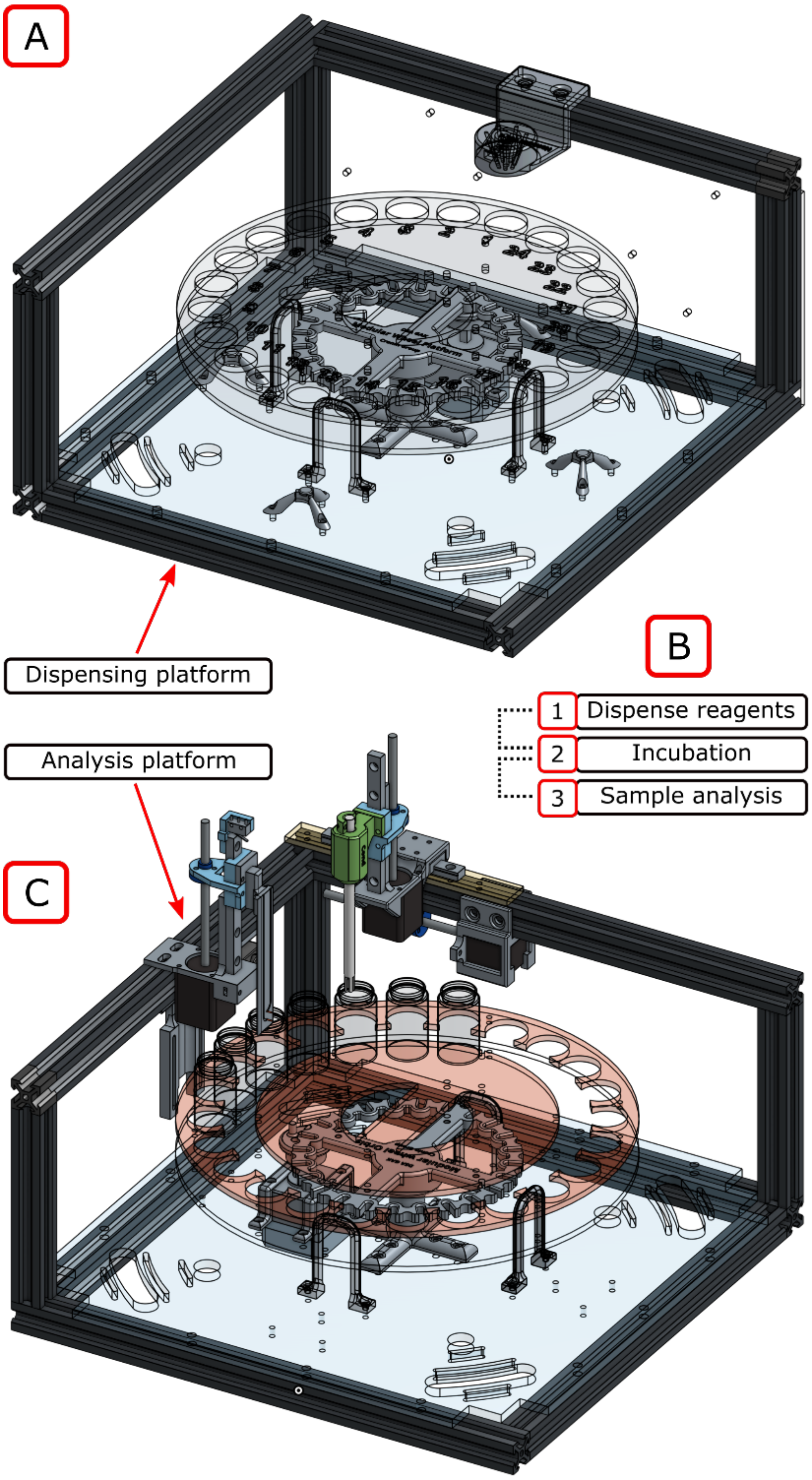


Fig. 2.3 Hardware of the two robotic platform: [A] Liquid handling platform R1. [B] Three step sequence of each reaction: Dispensing, sample incubation, sample analysis. [C] Platform R2 for reaction solution analysis.

by hand to the incubation chamber for 2 h at 50°C and 300 rpm to obtain the processed formulation. Sample are then added to R2 (Fig. 2.3C) for turbidity and pH analysis. As turbidity is a non-invasive technique in this it was performed first. Fig. 2.4 shows a CAD image of the two modules positioned around the platform frame, housing the turbidity and pH sensors (A and B respectively).

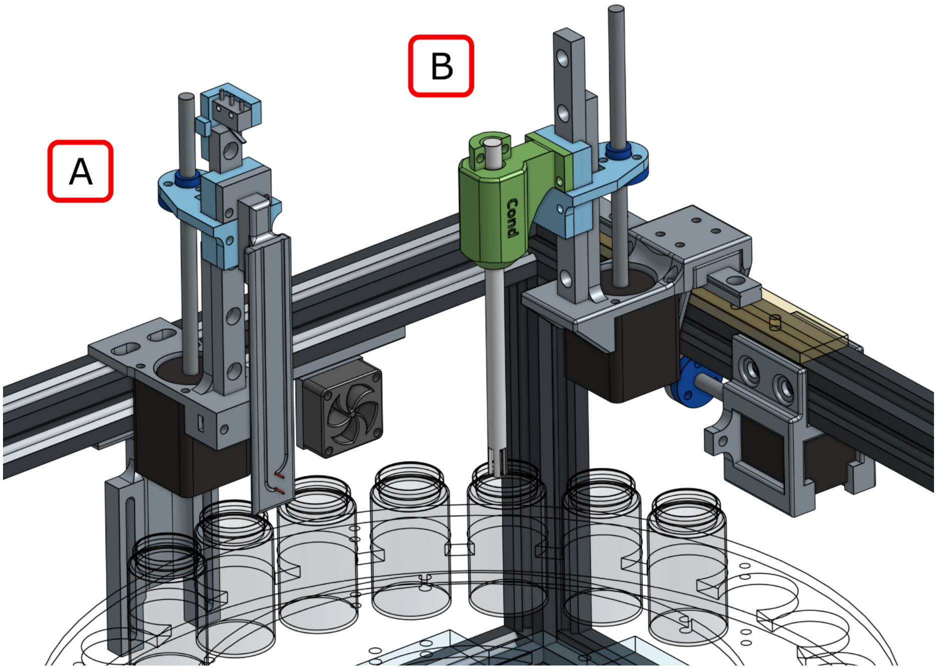


Fig. 2.4 CAD design of turbidity test module and pH test module: [A] Modular syringe driver for turbidity sensor. [B] Modular syringe driver for pH probe.

The pH probe used was a VWR TM 6mm x 150mm pH electrode with semi-micro epoxy body calibrated by using three buffers at know pH = 4, 7, and 10, respectively. The pH probe itself was mounted on a X-Z motion module. The purpose of this module is to allow pH analysis as well as the ability to run automated cleaning cycles on the probe between sample. The probe is removed from the reaction vial and moved to a static cleaning position (excluded from Fig. 2.4 for clarity, shown in detail below in Fig. 2.5). This position houses a 14 mL vial containing wash solution materials and is replenished after each cycle.

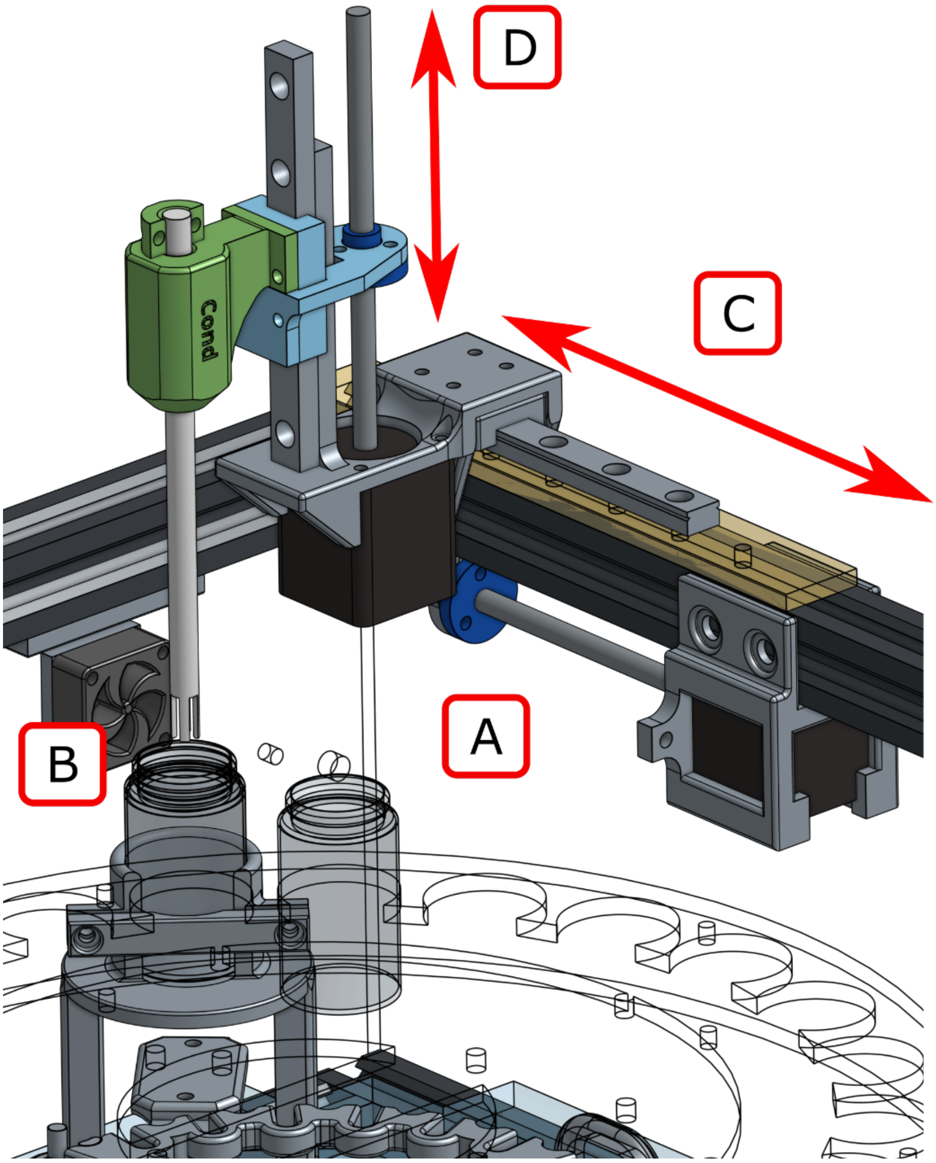


Fig. 2.5 Detailed view of pH probe set-up and motion. [A] Reaction/sample vial [B] Cleaning station vial [C] Horizontal movement [D] Vertical movement.

The procedure and method for determining sample turbidity is described in Section 2.2.3 below, this section describes the hardware set-up. Fig. 2.6 shows a CAD image of the light source and sensor for the turbidity measurement mounted on a modular syringe driver. The vial tray was cut to allow for the light source and sensor to be positioned directly opposite each other, passing through each vial. A background 3D print was also installed on the vial tray to provide a stable image background.

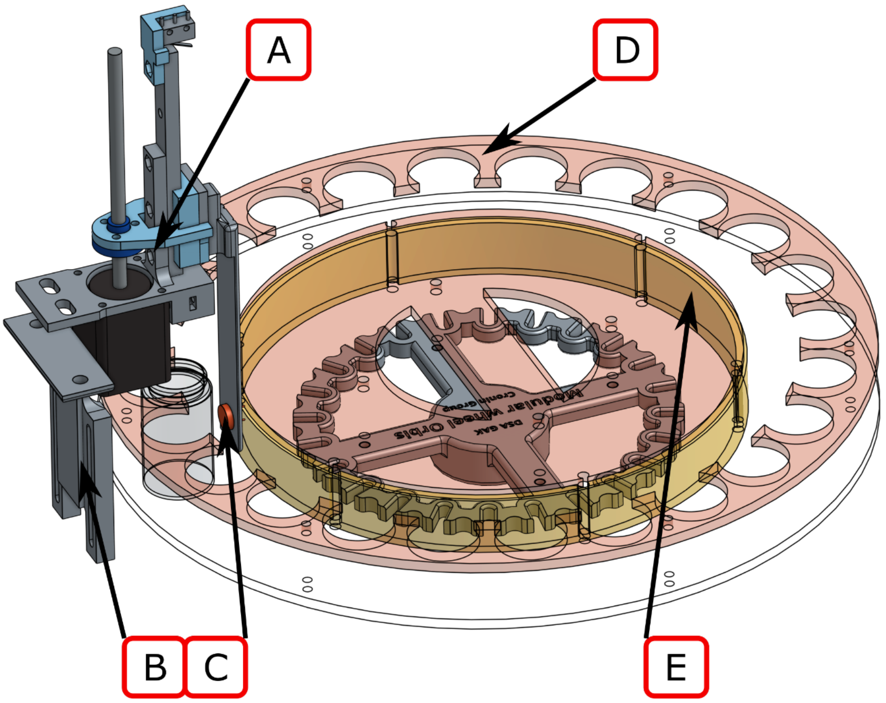


Fig. 2.6 CAD design of the light source and sensor for the turbidity measurement. [A] Static module syringe driver holding the turbidity sensor. [B] 3D printed mount for the light source, secured to the underside of the frame [C] Turbidity sensor [D] Custom vial tray to allow for sensor placement behind the vial [E] Imaging background embedded in the vial tray.

Turbidity sensor

Turbidity is an optical property based on the amount of light scattered and absorbed by colloidal and suspended particles [153]. Here we use the correlation between turbidity and the transition light intensity to measure the turbidity of the formulation product. A schematic layout and electronic set-up of the device are shown in Fig. 2.7. A moving LED light source and a light detector could move to the opposite sides of the sample vials. The value recorded by the light detector was converted into an electric power output proportional to the intensity of the light and converted to the turbidity value in NTU, through a calibration curve. The calibration curve was obtained by using turbidity standards (1, 2, 5, 10, 100, 500, and 1000 NTU).

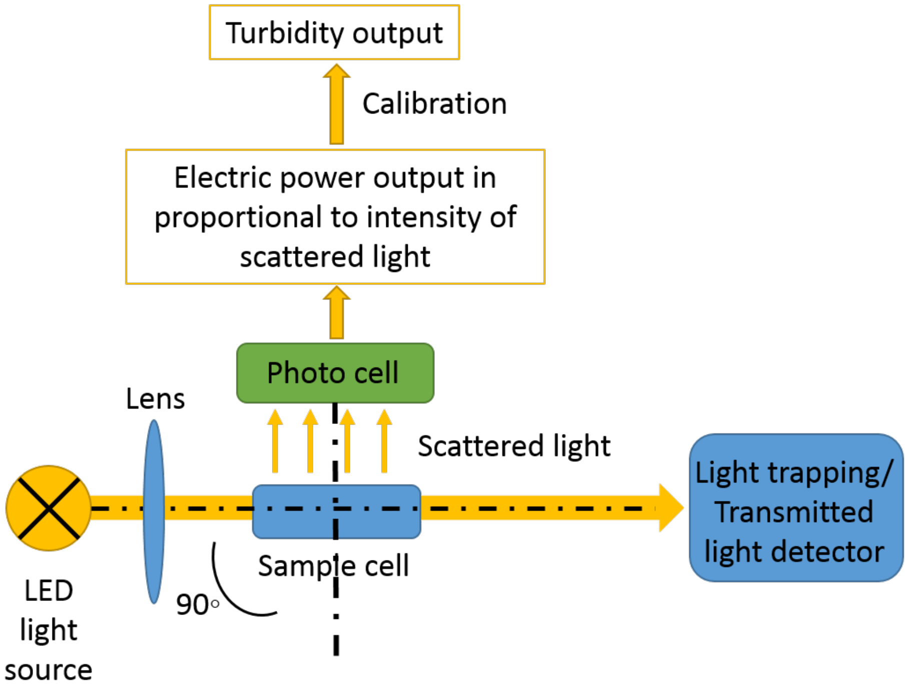

Turbidity output
Calibration
Electric power output in proportional to intensity of scattered light
Photo cell
Lens
Scattered light
Light trapping/ Transmitted light detector
Sample cell
90°
LED light source

Fig. 2.7 Schematic figure of the procedure and method for determining sample turbidity.

Viscosity measurement

The viscosity was tested off-line to measure the viscosity at a shear rate of 10 s^{-1} and 25 °C, by using a rotational viscometer (ARES Rheometric Scientific, strain controlled, couette configuration).

2.2.4 Robotic experiments

Preliminary operations

Surfactants were manually diluted with water to have an initial concentration of 20 g of active matter per litre and added to the feeding bottles. The thickener and the polymer were used pure. pH of all ingredients was manually pre-adjusted to 5.5 using citric acid and NaOH.

Sequence of operations

- A .csv file generated from the previous iteration of the algorithm is used to trigger the sample preparation in the first platform R1. The .csv file contains the amount of ingredients for each sample. A .txt file is generated at the same time containing the same information.
- Samples are transferred into the incubator to process the samples.
- Samples are cooled down to room temperature and transferred to the second platform.
- Turbidity of the samples is tested and the data is associated to the corresponding sample in the .txt file.
- Off-line viscosity tests are carried out and the data are added to the .txt file.
- The price is calculated based on the unitary price and the relative amounts of the adopted ingredients and added to the data file.
- The new collected data are merged with the ones obtained in all previous iterations.
- TS-EMO is run to generate new conditions to be tested.
- The new generated conditions are tested in silico using the stability classifier. The samples predicted to be unstable are discarder and the TS-EMO is run again until an

entire set of 8 samples is generated. This are summarized in a csv. file to trigger a new iteration.

Side tests operation

- The prepared samples were randomly tested for their pH, for some of the iterations, to confirm no pH variations occur during the preparation and processing. pH was only tested for some of the batches to save time and resources, since no pH variation was observed after processing in any of the investigated samples and preliminary observations.
- Pictures of the samples are taken to visualize phase separation. This is currently done by a human operator. Automated image analysis can be integrated in the future.

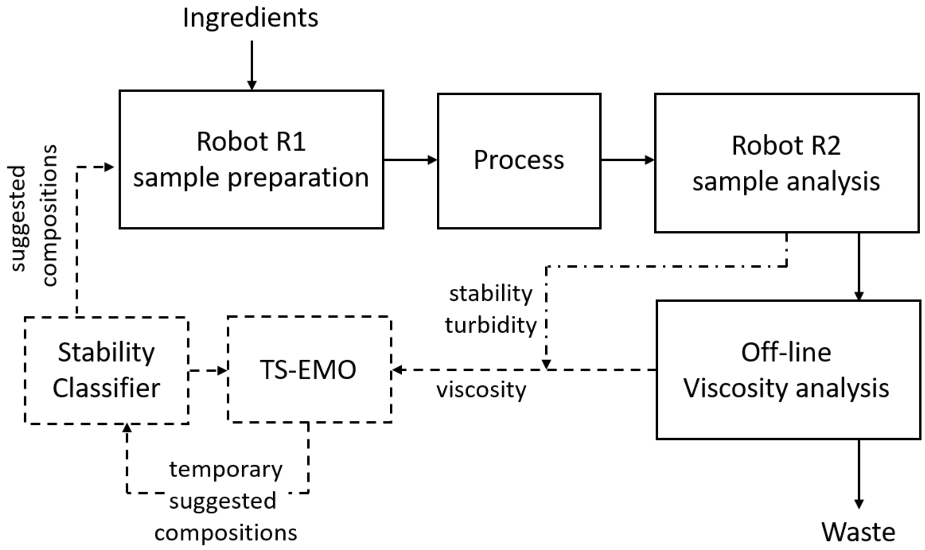


Fig. 2.8 Schematic diagramme of the sequence of operations.

2.2.5 Classification algorithm

Classification is a type of model assigning labels to regions of the parameters space given only a few known labelled training data [154]. The algorithm used in this study was developed according to the multiple active learning methodologies, developed and successfully applied to images classification [155], music annotation [156], and text categorization [157]. The method was applied to our case study to distinguish between stable and non-stable formulations faster than randomly accumulating and searching experimental evidence. A Naïve Bayes Classifier and an uncertainty-based sampling strategy were adopted. A flowchart for this framework is shown in Fig. 2.9.

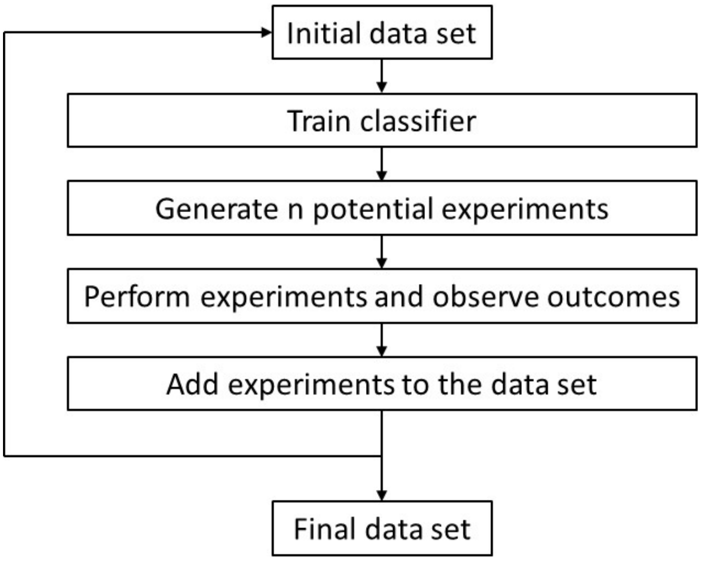

Initial data set
Train classifier
Generate n potential experiments
Perform experiments and observe outcomes
Add experiments to the data set
Final data set

Fig. 2.9 A workflow for active learning of the developed stability classifier.

A small set of initial data is needed to first train the model, and generate possible experiments for the next step. The trained classifier then can predict the outcome of these experiments and selects the most uncertain experiment, i.e. the one with the lowest confidence on the predictions. The selected experiment is then performed on the real system, and the result is added to the dataset, which can be used to train a new classifier. The process is repeated again until a given termination criterion is met. The so collected final dataset should be more informative than one built using a non-active acquisition method. In this work, a batch sequential design was used, suggesting 12 different experiments at each iteration.

According to Bayes theorem, the probability of given x being class c is Eq. (2.1) [158], with the assumption that all attributes are independent given the value of the class variables, Eq. (2.2).

$$p(y = c \mid \mathbf{x}) = \frac{p(\mathbf{x} \mid y = c)p(c)}{p(\mathbf{x})} \tag{2.1}$$

$$p(y = c \mid \mathbf{x}) \propto p(\mathbf{x} \mid y = c)p(c) \tag{2.2}$$

Due to the fact that features S1, S2 and S3 sum to a constant, Eq. (2.2) was modified to consider the subsets of independent features, following a joint distribution. The posterior distribution will then be given by Eq. (2.3).

$$p(y = c \mid \mathbf{x}) \propto p(c) \prod_{j} p(x_j \mid y = c) \tag{2.3}$$

where j denotes each subset.

In this work, there are three subsets of features: (i) features S1, S2 and S3 sum to a constant value, and follow the Dirichlet distribution, which is denoted as x_0,

$$p(x'_0 \mid y = c) = Dir(x'_0 \mid \alpha_c) \tag{2.4}$$

where $x'_0 = x_0/t$ and $\alpha_c \in R^3$ are the parameters of the Dirichlet distribution; (ii, iii) features P1 and T1 follow a normal distribution, which is denoted as x_i,

$$p(x_i \mid y = c) = N(x_i \mid \mu_{i,c}, \sigma_{i,c}) \tag{2.5}$$

where $\mu_{1,c}$ represents the mean of feature x_i prediction $y = c$, $\sigma_{1,c}$ represents the standard deviation of feature x_i for prediction $y = c$. $i = 1$ for P1, and $i = 2$ for T1.

Therefore, the posterior probability is given by

$$p(y = c \mid x) \propto p(\mathbf{x})p(x_0' \mid y = c)p(x_1 \mid y = c)p(x_2 \mid y = c) \tag{2.6}$$

Adopting a constant prior $p(\mathbf{x})$, Eq. (2.6) becomes:

$$p(y = c \mid x) \propto p(x_0' \mid y = c)p(x_1 \mid y = c)p(x_2 \mid y = c) \tag{2.7}$$

The parameters were estimated by using maximum likelihood estimation:

$$\max_{\theta} \log p(y = c \mid \mathbf{x}) \tag{2.8}$$

where $\theta = (\alpha_c, \mu_c, \sigma_c)$, μ_c and σ_C have analytical solutions by setting the derivation of log likelihood equals0; α_c is found by using the mean precision algorithm [159].

Given the nature of the classification algorithm, each batch at a given iteration, would likely be made of similar experiments. Therefore, the algorithm assigns a score to represent the importance of each sample in terms of local uncertainty and global exploration. From the score, we assign a probability to each of the sample according to a pair of pre-defined probabilities: local uncertainty and global exploration.

The local uncertainty is measured by Shannon entropy [130]:

$$S_{i,local} = \sum_{k \in c} \widehat{p_{i,k}} \log \widehat{p_i, k} \tag{2.9}$$

where k represents class k, $\widehat{p_i, k}$ is the predicted probability, and c is the number of the classes.

The score for global exploration is given by:

$$S_{i,global} = \min_j \left\|x_i - x_j\right\| \tag{2.10}$$

where j represent sample j. Using $S_{i,local}$ and $S_{i,global}$ we sort the samples separately according to each of scoring criteria. Then we assign the probability for each of the data using a discrete exponential distribution Eqs. (2.11) to (2.16). The parameter (λ) was set with the objective defined in such a way that top 5% of the experiments should be assigned 95% of the probability. In another words, we have 95% chance to sample a point within the 5% best points ranked by uncertainty value.

$$\mathbf{I}_{local} = (I_{1,local}, I_{1,local}, \cdots, I_{n,local}) \tag{2.11}$$

$$\mathbf{I}_{global} = (I_{1,global}, I_{1,global}, \cdots, I_{n,global}) \tag{2.12}$$

$$p_{i,local} = \frac{exp(index\ of\ i\ for\ i\ in\ \mathbf{I}_{local} \mid \lambda)}{\sum_{j=1}^{n} exp(index\ of\ j\ for\ j\ in\ \mathbf{I}_{local}) \mid \lambda} \tag{2.13}$$

$$p_{i,global} = \frac{exp(index\ of\ i\ for\ i\ in\ \mathbf{I}_{global} \mid \lambda)}{\sum_{j=1}^{n} exp(index\ of\ j\ for\ j\ in\ \mathbf{I}_{global}) \mid \lambda} \tag{2.14}$$

$$\mathbf{p}_{local} = (p_{1,local}, p_{2,local}, \cdots, p_{n,local}) \tag{2.15}$$

$$\mathbf{p}_{global} = (p_{1,global}, p_{2,global}, \cdots, p_{n,global}) \tag{2.16}$$

where $\mathbf{p}_{local}$ and $\mathbf{p}_{global}$ are arrays of assigned probability according to S_{local} and S_{global}, and i is used to denote the index of the sample. Therefore, the overall probability for sampling data is given by Eq. (2.17).

$$\mathbf{P}'_{overall} = \mathbf{p}_{local} \odot \mathbf{p}_{global} \tag{2.17}$$

where $\odot$ is element-wise multiplication.

The samples are sorted according to the values of $\mathbf{P}'_{overall}$, and the overall probability for sampling is given by Eq. (2.19).

$$\mathbf{I}_{overall} = (I_{1,overall}, I_{1,overall}, \cdots, I_{n,overall}) \tag{2.18}$$

$$p_{i,overall} = \frac{exp(index\ of\ i\ for\ i\ in\ \mathbf{I}_{overall} \mid \lambda)}{\sum_{j=1}^{n} exp(index\ of\ j\ for\ j\ in\ \mathbf{I}_{overall}) \mid \lambda} \tag{2.19}$$

$$\mathbf{p}_{overall} = (p_{1,overall}, p_{2,overall}, \cdots, p_{n,overall}) \tag{2.20}$$

2.2.6 Multi-objective optimization algorithm

Here we adopted the Thompson Sampling Efficient multiobjective optimization (TSEMO) algorithm [146], which is extended from the Thompson Sampling (TS) method from the multi-arm bandit community to continuous multi-objective optimization.

Objective function

For a multi-objective optimization problem, it can be defined as

$$min_{x \in X \subset \mathscr{R}^d} \mathbf{G}(\mathbf{x}) = [g_1(\mathbf{x}), g_2(\mathbf{x}), \cdots, g_m(\mathbf{x})] \tag{2.21}$$

where X is the design space, **x** is the decision vector and **G** is a vector of m scale objectives $g_i(\mathbf{x})$ to be minimized.

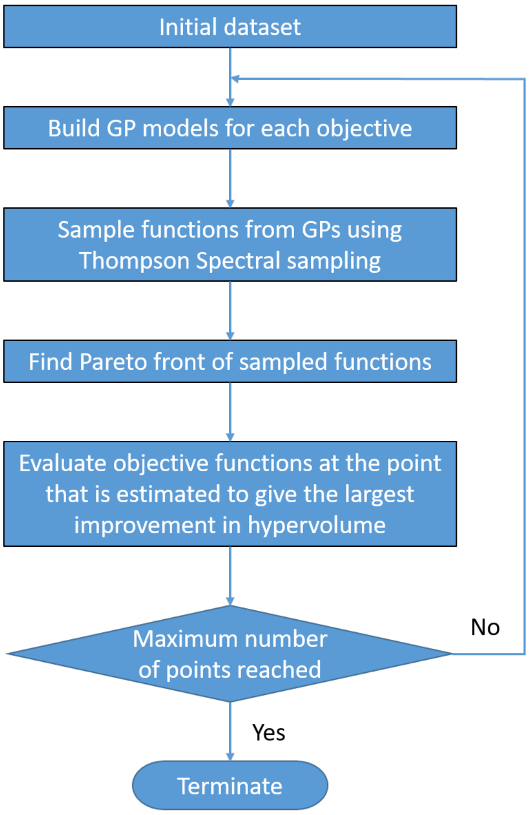

Initial dataset
Build GP models for each objective
Sample functions from GPs using Thompson Spectral sampling
Find Pareto front of sampled functions
Evaluate objective functions at the point that is estimated to give the largest improvement in hypervolume
Maximum number of points reached
No
Yes
Terminate

Fig. 2.10 Algorithm flowchart for TESMO algorithm.

Algorithm outline

The outline of the TSEMO algorithm is shown in the flowchart below. As indicated in Fig. 2.10 the initial Gaussian Process models will be built with the initial dataset with size n. At each iteration i, Let $X^{(i)} = \mathbf{x}_1, \cdots, \mathbf{x}_n$, $\mathbf{x}_{n+1}, \cdots, \mathbf{x}_{n+i}$ be the inputs of the data collected, and $Y_j^{(i)} = y_j^{(1)}, \cdots y_j^{(n)}$, $y_j^{(n+1)}, \cdots, y_j^{(n+i)}$ the corresponding responses for each objective function $g_j(\mathbf{x})$, with $j = 1, \cdots, m$. For each objective $Y_j^{(i)}$, a corresponding independent GP is trained, that we find $GP_j^{(i)}(m^{(i)}, k^{(i)} \mid X^{(i)}, Y_j^{(i)})$ for $j = 1, \cdots, m$. For GP regression, covariance functions, also known as the kernel function, determines the properties of the fitted functions. In this work, we used stationary kernel functions from the Matérn class. The kernel is given by Eq. (2.22) [160].

$$k_{Matrn}(x_i,x_j) = \frac{2^{1-\nu}}{\Gamma(\nu)}\left(\frac{\sqrt{2\nu}r}{l}\right)^{\nu} K_\nu\left(\frac{\sqrt{2\nu}r}{l}\right) \tag{2.22}$$

where $r = \sqrt{(x_i - x_j)\Lambda(x_i - x_j)}$, $\Lambda = diag\left(\lambda_1^{-2}, \cdots, \lambda_d^{-2}\right)$, $K_\nu(\cdot)$ is a modified Bessel function and $\Gamma(\cdot)$ is the gamma function. The parameter ν defines the smoothness of Matérn covariance functions. The parameters λ_i define the length scales of the input variables. If an unput dimension is not relevant, the corresponding length scale is large. The maximum a posteriori extimate (MAP) method was used to train the hyperparameters.The MAP likelihood is then given as:

$$\begin{aligned} L_{MAP}(\xi) = & -\frac{1}{2}log(|\Sigma|) - \frac{1}{2}y^T\Sigma^{-1}y - \frac{n}{2}log(2\pi) \\ & + \sum_i\left(-\frac{1}{2}log(2\pi) - \frac{1}{2}log(\sigma_i^2) - \frac{1}{2\sigma_i^2}(\xi_i - \mu_i)^2\right) \end{aligned} \tag{2.23}$$

where μ_i and σ_i^2 denote the mean and the variance of the normal distribution of the prior. The MAP hyperparameter estimate is then given by the following optimization problem:

$$\xi_{MAP} \in arg \max_{\xi} L_{MAP}(\xi) \tag{2.24}$$

After the GPs are trained, for each objective, single-objective Thompson Sampling (TS) were applied to have m distinct functions from these m independent GPs using spectral sampling. From this, we obtain a collection of m functions $f_1^{(i)}(\mathbf{x}), \cdots, f_m^{(i)}(\mathbf{x})$ at each iteration. Since the GP samples are cheap-to-evalute functions, we then applied NSGA-II, a genetic multi-objective algorithm. Let $C^{(i)}$ refer to the current candidate set given by the approximate Pareto set of GP samples $f_1^{(i)}(\mathbf{x}), \cdots, f_m^{(i)}(\mathbf{x})$. The size of the candidate set $C^{(i)}$ is equal to the population size, and the number of generations is fixed to allow sufficient convergence of this set to the true Pareto set the GP samples $f_1^i(\mathbf{x}),\cdots,f_m^{(i)}(\mathbf{x})$.

Once we have the approximate Pareto set $C^{(i)}$, hypervolume criterion was used to choose the next sampling point. The hypervolume indicator is the m-dimensional lebesgue measure λ_m of a dominated subspace limited above by a reference point (Eq. (2.25)).

$$HV(\mathbb{P},\mathbf{R}) = \lambda_m \left(\bigcup_{p \in \mathbb{P}} [\mathbf{p},\mathbf{R}] \right) \tag{2.25}$$

where $\mathbb{P}$ is the non-dominated Pareto front, and $\mathbf{R}$ is the reference point.

The point to sample should give the largest hypervolume improvement when it add to the current Pareto set (Eq. (2.25)).

$$\mathbf{x}_{n+i+1} \in arg \max_{x \in C^{(i)}} \Delta HV \left(\mathbf{y}_C, \mathscr{P}^{(i)}, \mathbf{r}^{(i)} \right) \tag{2.26}$$

where $\mathscr{P}^{(i)}$ is the Pareto front of the current output dataset, $\mathbf{r}^{(i)}$ is the reference point for the hypervolume calculation, $\mathbf{y}_C = \left(f_1^{(i)}(\mathbf{x}),\cdots,f_n^{(i)}(\mathbf{x}) \right)$, and

$$\max_{x \in C^{(i)}} \Delta HV \left(\mathbf{y}_C, \mathscr{P}^{(i)}, \mathbf{r}^{(i)} \right) = HV \left(\mathscr{P}^{(i)} \cup \mathbf{y}_C, \mathbf{r}^{(i)} \right) - HV \left(\mathscr{P}^{(i)}, \mathbf{r}^{(i)} \right)$$

It is important to have a fast algorithm to calculate δHV due to the fact that this calculation needs to be done for every candidate point in $C^{(1)}$ A summary for the efficient computation of ΔHV is given by Emmerich et al. [161]. Due to the fact that we have three objective, therefore we use a Monte-Carlo approximation for the calculation of ΔHV.

The reference point $\mathbf{r}^{(i)}$ is approximated by the anti-ideal point of approximate Pareto front of GP samples $\left\{f_1^{(i)}(\mathbf{x}),\cdots,f_n^{(i)}(\mathbf{x})\right\}$ from the candidate set $C^{(i)}$:

$$\mathbf{r}^{(i)} = \left(\max_{x\in C^{(i)}}\left(f_1^{(i)}(\mathbf{x})\right),\cdots,\left(f_m^{(i)}(\mathbf{x})\right)\right) \tag{2.27}$$

Lastly, the datasets are updated with the proposed data point: $X^{(i+1)} := \mathbf{x}_1,\cdots,\mathbf{x}_n,\mathbf{x}_{n+1}$, $\mathbf{x}_{n+i},\mathbf{x}_{n+i+1}$ and $Y^{(i)}{}_j := \left\{y_j^{(1)},\cdots,y_j^{(n)},y_j^{(n+1)},\cdots y_j^{(n+1)}\right\}$ for $j=1,\cdots,m$ and the procedure is repeated with $i := i+1$ until a the maximum number reached.

Batch sampling

Due to the fact that in this case we can run up to 24 samples at each iteration, it is advantageous to adopt to batch-sequential design in order to propose multiple sampling points. Let b denote the number of sampling points and i the current iteration. TSEMO was employed i-times to add $i\times b$ number of points. The procedure is same as shown in Section 2.2.6 using the data $X^{(i)}$ and Y_j^i except that Eq. (2.26) is replaced with the equation which can propose multiple sampling points. Then a greedy approximation was used for the optimization. The multiple sampling points at each iteration can be found in Eq. (2.28)

$$\begin{aligned}
&\mathbf{x}_{n+i\times b+1} \in arg \max_{x\in C^{(i)}} \Delta HV(\mathbf{y}_C,\mathscr{P}^{(i)},\mathbf{r}^{(i)}) \\
&\mathbf{x}_{n+i\times b+2} \in \\
&arg \max_{x\in C^{(i)}} \Delta HV\left(y_C,\mathscr{P}^{(i)}\bigcup\left\{\left(f_1^{(i)}(x_{n+1\times b+1},\cdots,f_m^{(i)}(x_{n+1\times b+1})\right)\right\},r^{(i)}\right) \\
&\qquad\qquad\vdots \\
&x_{n+i\times b+b} \in arg \max_{x\in C^{(i)}} \Delta HV\left(\mathbf{y}_C,\mathscr{P}^{(i)})\cup\left\{\left(f_1^{(i)}(x_{n+i\times b+1})\right),\cdots,\left(f_m^{(i)}(x_{n+i\times b+1})\right)\right\},\right) \\
&\qquad\cup\left\{\left(f_1^{(i)}(x_{n+i\times b+2})\right),\cdots,\left(f_m^{(i)}(x_{n+i\times b+2})\right)\right\} \\
&\qquad\cup\cdots\left\{\left(f_1^{(i)}(x_{n+i\times b+b-1})\right),\cdots,\left(f_m^{(i)}(x_{n+i\times b+b-1})\right)\right\}
\end{aligned} \tag{2.28}$$

where $\mathbf{y}_C = \left(f_1^{(i)}, \cdots, f_m^{(i)}\right)$.

The optimization procedures are carried out are similar to Section 2.2.6. The hypervolume for all points in $C^{(i)}$ were calculated and the datasets are updated with the proposed data sampled points:$X^{(i+1)} := \mathbf{x}_1, \cdots, \mathbf{x}_n, \mathbf{x}_{n+1}, \cdots, \mathbf{x}_{n+i\times b}, \mathbf{x}_{n+i\times b+a}, \cdots, \mathbf{x}_{n+i\times b+b}$ and $Y_j^{(i)} := \left\{Y_j^{(1)}, \cdots, Y_j^{(n)}, Y_j^{(n+1)}, \cdots, Y_j^{(n+i\times b)}, g_j(\mathbf{x}_{n+i\times b+1}), \cdots, g_j(\mathbf{x}_{n+i\times b+b})\right\}$ for $j = 1, \cdots, m$.The procedure will be repeated with $i := i+1$ until a specified maximum number of function evaluations are reached. Further discussion of the TSEMO algorithm and be found in Ref. [146].

Implementation of TSEMO algorithm

As shown in Section 2.2.6, the optimization procedure can be stopped when the maximum number of evaluations is reached or when the operator is satisfied with the obtained results. This can be automated by terminating the algorithm when the objective functions are lower than a given epsilon. For this specific case study, the number of suggested experiments at each iteration was set equal to 8. The input variables were chosen as the concentrations of the ingredients and the three targets to minimise were chosen as the turbidity value in NTU, the squared distance between the measured viscosity and the target viscosity of 3 $Pa \cdot s$, and the cost of the adopted ingredients ($\$ \cdot L^{-1}$). The latter was calculated as the sum as the unitary prices of the ingredients multiplied by the adopted amounts in each sample. As this target was an explicit function of the input variables, the code was modified to not train a GP for this specific target, using the directly calculated values instead. The targets were chosen according to the indications of the product supplier: an ideal product is a stable homogeneous clear formulation with a viscosity close to 3 $Pa \cdot s$, at the lowest possible cost.

2.3 Results and discussion

2.3.1 Stability prediction: classification algorithm

A classification algorithm was developed and trained in order to classify the suggestions of the coupled TS-EMO algorithm based on their stability, and to avoid waste of resources and time generating samples the exhibit the undesired phase separation. The smart sampling classifier was developed as described in Section 2.2.5. An initial data set of 48 samples was generated using a Latin Hypercube Design as a space filling technique [162]. The samples were generated by two complete rounds of the robotic platforms R1 and R2. The initial data set was used to train the classifier and generate 12 new potential experiments to be performed at each iteration. The new generated samples were added to the data set and procedure repeated for four complete cycles. As described in detail in Section 2.2.5, a Naïve Bayes Classifier was chosen due to its simplicity, efficiency and accuracy in classification problems. The performance was evaluated as the average for prediction accuracies of the 5-fold cross-validation between the classes of stable and unstable formulations [163]. A bi-dimensional representation of the initial data set is shown in Fig. 2.11 (Run1), together with the first 12 suggested experiments. Linear Discriminant Analysis (LDA) was found to be a suitable dimensionality reduction algorithm for better visualization of the data set. The results of the iterative training of the classifier are shown in Fig. 2.11 (Smart sampling). As one can see, the suggested experiments at each iteration are nicely distributed at the border between the two clusters, which represents the area with the highest uncertainties. Moreover, the repulsive criterion adopted for the batch sequential design, provided a better exploration of the space. For the sake of comparison, the same procedure was repeated adopting a random sampling strategy. For the sake of comparison, representation of the suggested experiments using a random sampling and smart sampling strategy is shown in Fig. 2.11.

The prediction accuracy was evaluated using a Supporting Vector Machine (SVM)

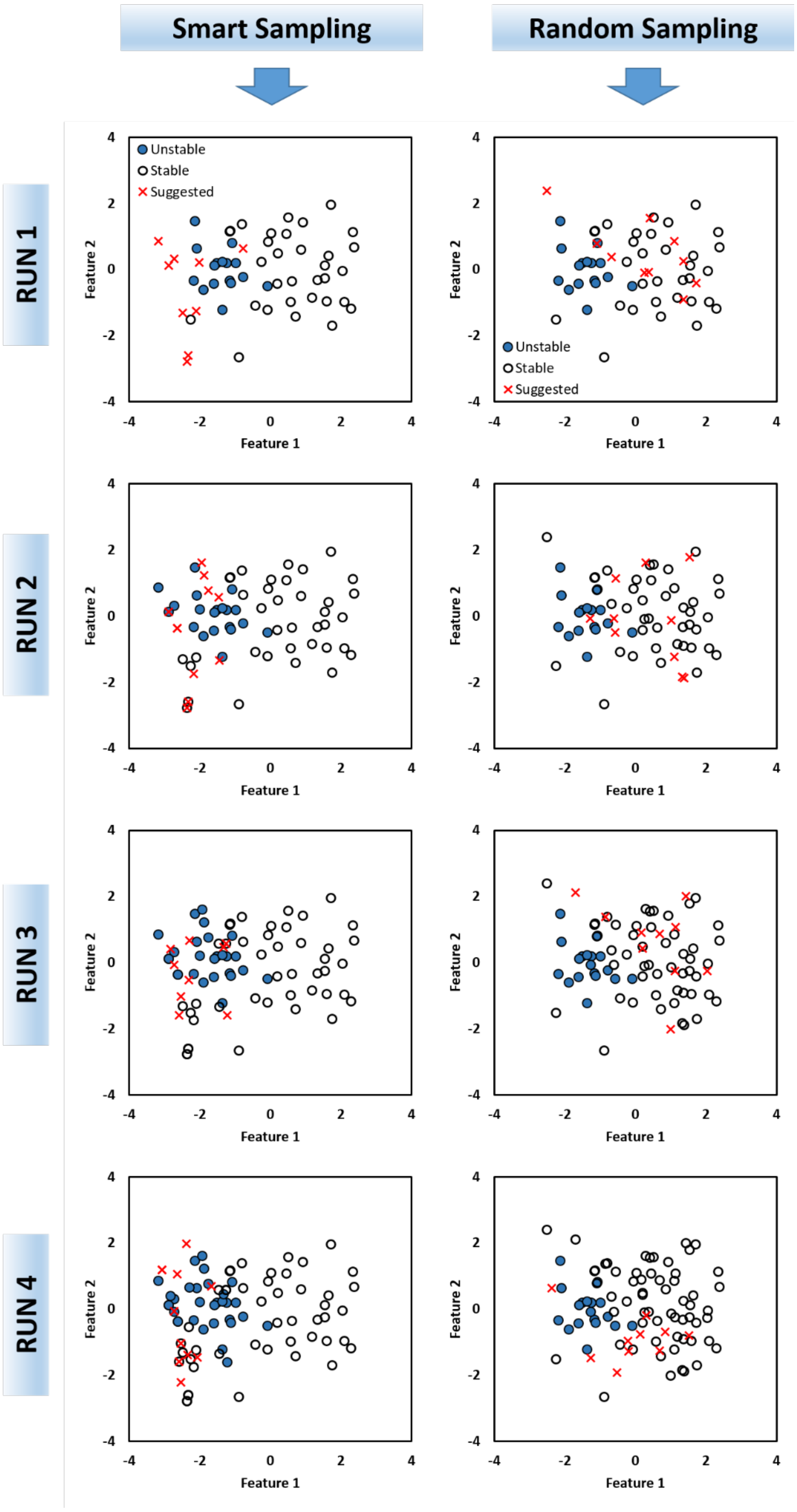


Fig. 2.11 2D visualization of the comparison between smart sampling and random sampling.The circles represent the stable samples. The circles filled in blue represent the unstable ones. The red crosses are the suggested experiments to be run. It can be seen that compared to the random sampling method, by applying smart sampling algorithm, the suggested experiments to be run are mostly located at the boundary part of the stable area.

classifier. In fact, as a Naïve Bayes model was used in the active learning algorithm method, therefore we might have collected data solely tailored to the model build by the naïve Bayes classifier. Fig. 2.12 shows the prediction accuracy of the active learning algorithm and the random sampling, proving the superiority of the former, at each iteration.

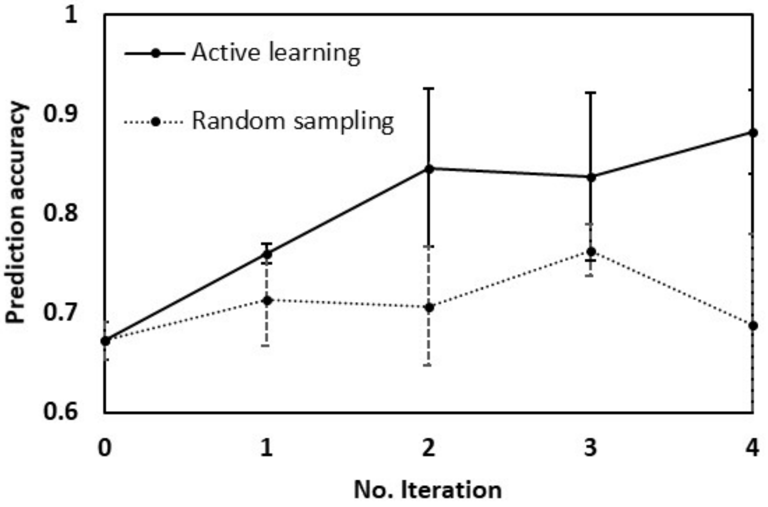


Fig. 2.12 Prediction accuracy at different iterations using active learning and random sampling.

2.3.2 Optimization results

The same 96 data points collected to train the classification algorithm described in Section 2.3.1 were used to initiate the TS-EMO algorithm. Once initiated, 16 iterations of the optimization procedure were carried out generating a total of 128 samples within 15 working days. As previously described, the algorithm suggests conditions in order to find the best predictions of the actual Pareto front, minimizing uncertainties and finding a compromise between the minimization of the conflicting objective functions. The target properties for the specific product under considerations were (i) stability and low turbidity, (ii) honey-like viscosity (target 3 $Pa \cdot s$ at a shear rate of 10 s^{-1} and 25 $°C$), and (iii) low price of the adopted ingredients. Interestingly, the percentage of suggested unstable formulations, presenting phase separation significantly decreased over the iterative optimization loop and the algorithm stopped suggesting unstable conditions from the 12th iteration, as shown in Fig. 2.13. This can be ascribed to the fact that unstable samples often present a higher value of turbidity which is one the objective chosen for the optimization procedure. However, the integration of the stability classifier helped to avoid running experiments giving unstable products, helping to save time and reducing the waste of resources.

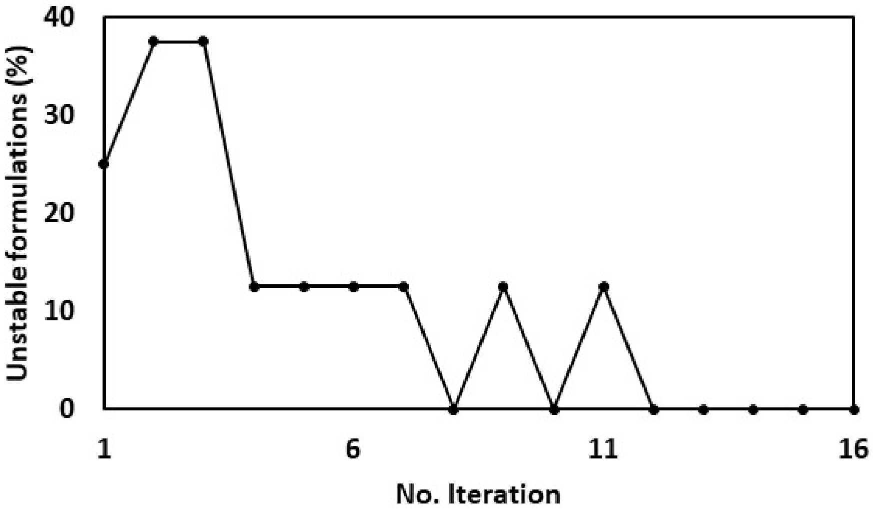


Fig. 2.13 Percentage of suggested unstable formulations at each iteration.

The experimentally collected data were automatically analysed to provide a full list of the non-dominated experimental solutions, which represent the experimental Pareto front of the data set. Non-dominated solutions were defined as the ones where an improvement in one objective would lead to a worsening in at least one other objective. The full list of 32 non-dominated solutions identified, 11 of which in the training data set, and 19 in the suggested experiments, is provided in Table 2.1. In Table 2.2 we reported non-dominated solutions meeting the viscosity criterion. The full data set is reported in the Appendix.

As one can see, although a good number of clear formulations was already present in the training set, the algorithm was able to explore the input space more efficiently finding alternative solutions with a significant reduction in the price. The obtained solution was

Table 2.1 Result comparison; Non dominated solutions.

	S1	S2	S3	P1	T1	Turbidity	Viscosity	Price
	g L^{-1}	g L^{-1}	g L^{-1}	g L^{-1}	g L^{-1}	NTU	$mPa \cdot s$	\$ L^{-1}
Training set	2.51	4.16	8.33	0.22	0.51	1.6	16	1.66
	3.97	3.16	7.87	0.84	1.6	11.2	2678	2
	0.1	8.24	6.67	0.18	0.15	3.3	2	1.38
	4.61	2.62	7.76	1.24	1.04	4.4	3574	2.06
	2.59	5.98	6.44	0.38	0.82	11.7	855	1.73
	0.73	5.4	8.87	1.84	1.44	449.7	2595	1.79
	1.81	5.85	7.34	0.55	0.92	4	746	1.69
	8.4	3.18	3.43	0.48	1.8	17.2	2948	2.43
	2.79	2.93	9.28	1.72	0.88	7.4	2633	1.92
	6.16	4.29	4.55	0.52	1.77	56.1	2889	2.2
	10.94	3.94	0.12	1.8	1.36	28	2992	2.8
Suggested data	4.02	3.09	0.99	0.99	1.39	15.1	2824	2
	3.01	0.64	11.35	0.53	1.75	49.6	2789	1.87
	3.06	1.99	9.95	0.31	1.42	23.7	3301	1.82
	0.05	4.31	10.64	0.43	0.35	8.5	10	1.42
	0	0.02	14.98	0.01	0	18.3	54	1.32
	0	0.02	14.99	0	0	9.1	6	1.32
	1.05	3.24	10.7	0.31	0.52	18.2	88	1.52
	1.99	3.82	9.2	0.21	0.87	15.1	187	1.65
	0.46	0.46	14.08	0.14	0.59	17.4	24	1.44
	0.32	9.86	4.82	0.5	0.28	2.8	54	1.46
	0	0.27	14.73	0	0	18.4	4.1	1.32
	0.39	4.94	9.68	0.27	0.13	12.1	16	1.41
	0.06	1.89	13.05	0.95	0.77	26.3	22	1.53
	0	1.78	13.22	0.01	0.25	11.9	6	1.35
	0.7	13.74	0.56	0.45	0.25	23.7	4	1.49
	0.05	2.2	12.75	0.51	1.43	29.1	302	1.54
	0.06	12.16	2.78	0.02	1.99	16.3	1735	1.55
	0.14	3.69	11.17	0.31	0.96	187	1976	1.58
	0.27	5.2	9.53	0.79	0.1	34.2	111	1.47

Table 2.2 Non dominated solutions passing the viscosity criterion.

	S1	S2	S3	P1	T1	Turbidity	Viscosity	Price
	g L^{-1}	g L^{-1}	g L^{-1}	g L^{-1}	g L^{-1}	NTU	$mPa \cdot s$	\$ L^{-1}
Training set	3.97	3.16	7.87	0.84	1.6	11.2	2678	2
	4.61	2.62	7.76	1.24	1.04	4.4	3574	2.06
	8.4	3.18	3.43	0.48	1.8	17.2	2948	2.43
	2.79	2.93	9.28	1.72	0.88	7.4	2633	1.92
	6.16	4.29	4.55	0.52	1.77	56.1	2889	2.2
	10.94	3.94	0.12	1.8	1.36	28	2992	2.8
Suggested data	4.02	3.09	0.99	0.99	1.39	15.1	2824	2
	3.01	0.64	11.35	0.53	1.75	49.6	2789	1.87
	3.06	1.99	9.95	0.31	1.42	23.7	3301	1.82

compared with the best solution provided in a data set guided by experts' intuition. In this case the closest solution to the target was found using the following recipe: S1 = 4.00 g L^{-1}, S2 = 5.00 g L^{-1}, S3 = 6.00 g L^{-1}, P1 = 2.00 g L^{-1}, T1 = 2.00 g L^{-1}. In this case, a homogeneous formulation with a viscosity of 9270 mPa·s was obtained, with a turbidity value higher than 200 NTU and a cost of 2.19 \$ L^{-1}, proving that an appropriate space filling technique coupled with an algorithmic search can significantly outperform human intuition, in a relatively short amount of time and with very little prior knowledge about the physical system.

In order to evaluate the predictive capability of the trained surrogate models, the predicted Pareto front was plotted together with the experimental optima. In Fig. 2.14 a surface was fitted to the predicted non-dominated solutions and reported together with the data used for the initial training of the TS-EMO, all suggested conditions, and the non-dominated experimental optima. As shown, the predicted Pareto front gives a good approximation of the actual best solutions, laying in the neighbourhood of the calculated surface. *A posteriori* analyses of the shape of the Pareto front can also give some physical information to human operators to have a better insight of the system. From the data reported in Fig. 2.14. it is clear that, as a general trend, as the viscosity approaches the target value, the price and the turbidity of the system tend to increase. This would suggest that, on average, the most expensive ingredients (S2, T1, P1) would be the ones responsible for an increase in viscosity and turbidity of the samples, which was found to be the case for most of the samples in the data set. Of course, these are only general semi-quantitative indications, which do not reveal any information about more complex interactions that might occur between the different ingredients at different concentrations. However, it is worth noticing that the presented methodology can also offer some guiding lines to experts for further improvements and considerations about the actual physical role of the input variables on the desired properties of the product.

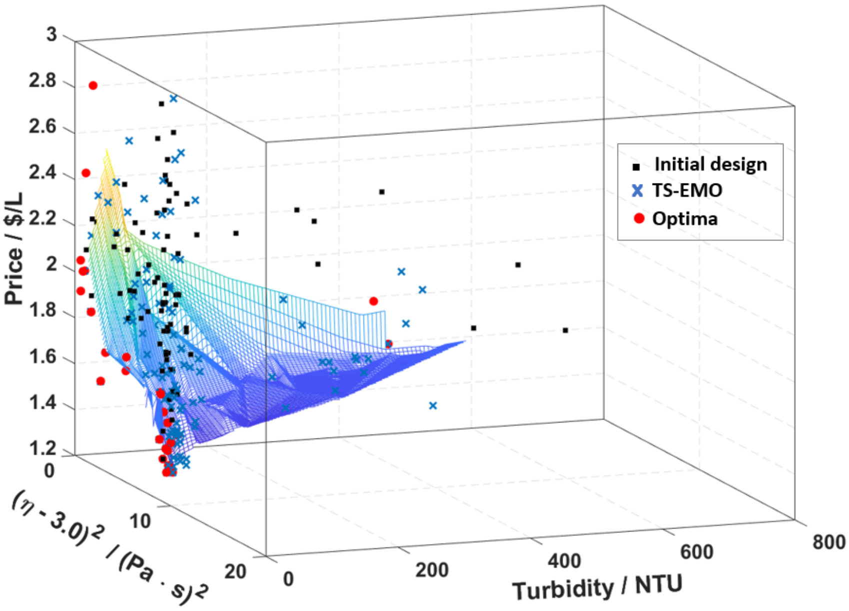


Fig. 2.14 Data set, experimental, and predicted Pareto front.

In this regard, the values of the hyperparameters of the trained GPs can also provide information about the relevance of the input variables for each objective function [10] . For the adopted surrogate models, a lower value of the length scale hyperparameter λ_i of an input variable indicates a greater contribution to the objective. The values of the length scale hyperparameters are reported in Table 2.3.

Table 2.3 The length scale hyperparameters of GP models.

	GP1 (viscosity)	GP2 (turbidity)
S1	$8.57 \cdot 10^{-2}$	$5.90 \cdot 10^{-2}$
S2	$3.16 \cdot 10^{-1}$	$1.62 \cdot 10^{-1}$
P1	$9.42 \cdot 10^{\,0}$	$7.58 \cdot 10^{-1}$
T1	$4.00 \cdot 10^{-2}$	$7.47 \cdot 10^{-2}$

The analysis of the hyperparameters suggests again a stronger influence of T1 on the viscosity and the turbidity; more complex interactions between S1 and T1 seem to be responsible for variations in the viscosity value, whereas S2 seem to also have a relevant effect on the turbidity of the samples. This kind of qualitative information may lay the foundation for integrated approaches for the simultaneous black-box optimization and physical knowledge generation by using robotic platform, in combination with other recently published promising methodologies [110, 164, 165].

Chapter 3

Surfactants selection and co-development of product and process of formulated products

3.1 Introduction

Fast development of formulations is of crucial importance in the chemical industry because of their ubiquity and the continuous need for new greener solutions. In Chapter 2, a closed-loop optimization scheme was presented in order to speed up the R&D process of a commercial liquid formulated product design. However, there are still quite a few open questions: i.e. the selection of M ingredients out of N potential compounds in the library, the co-development of formulation and the process condition.

In many formulated product design cases, formulation design is based on the choice of a certain subset of M components from a large number N of available chemicals (M < N). One typical example is the choice of a certain number of surfactants, which are used as stabilizers of emulsions and also influencing their final properties [166]. Moreover, the number of possible combinations is very large when many components are available. Due to manufacturing constraints or regulation issues, binary and ternary mixtures are often used in practice [167]. As a result, finding suitable binary or ternary mixtures with desired properties from all possible binary and ternary combinations is challenging [168–170]. For example, in a ternary mixture design (m = 3), if there are n =10, 20, or 50 possible components to choose from, there are 120, 1140, and 19600 combinations, respectively. As n gets larger, finding an optimal design for each possible combination and comparing the obtained results becomes increasingly prohibitive. The situation is complicated by the fact that each component can be used at different concentrations, corresponding to different final properties, further expanding the search space. Here, the research question is whether is it possible to explore the large design space efficiently and optimize for the desired properties under the constrains of the number of selected ingredients.

The integration of product and process design is also crucial, especially when the specific product microstructural attributes strongly depend on the selected manufacturing technologies and operating conditions are crucial [171, 172]. In 2004, Prof Grossmann introduced product-process design as one of the future challenges of chemical engineering [173]. Since then, various attempts have been made to develop systematic methodologies. Similar to product design only, there are mainly three approaches to the design: trial-and-error experimental approach, computational approach and hybrid experiment-and model-based technique. Here we would like to extend the previous approach presented in Chapter 2 and see whether it can help with the co-development of product and process design.

3.2 Material and Methods

3.2.1 Case study and materials

The case study under consideration in this work is the same commercial formulation presented in Chapter 2 but with extensions. In detail, five available surfactants are used: Dehyton PK 45 (S1), Dehyton AB 30 (S2), Plantacare 818 (S3), Plantacare 2000 (S4), and Texapon SB 3 (S5). One polymer (P1 = Dehyquart CC7), a thickener (T1 = Arlyon TT) were used. pH was adjusted using citric acid (ACS reagent, $\geq$ 99.5%, Sigma-Aldrich); used as received.

3.2.2 Experimental set-up

The experimental samples were generated using a previously developed semi-automated robotic platform. A detailed description of the robotic platform can be found in Chapter 2. Briefly, as shown in Fig. 3.1, the platform consists of two separated stations for the preparation and the analysis of the samples. The algorithmic procedure developed in this work generated a .csv file containing the experimental design to be tested. This triggered the first station consisting of 8 syringe pumps separately feeding the components to a dispensing element. This was used to fill a batch of up to 24 sampling vials (Vsample = 10 mL) allocated on a rotating wheel. All surfactants were previously diluted in water to achieve a concentration of 20 $g \cdot L^{-1}$ in the feeding bottles and the pH was adjusted to 5.5. The generated samples were transferred into an incubator (Corning LSE 71L Shaking Incubator) where they are shaken at a fixed temperature, mixing power, and time. The processed samples were cooled down to room temperature and placed on the rotating wheel of the second station where the samples were automatically processed through a camera to distinguish between stable (homogeneous) and unstable (presenting phase separation) formulations. Automatic pH tests were also carried out and no pH variations were observed in any sample after the processing.

Finally the viscosity of the samples at a shear rate of 10 s^{-1} was measured off-line using a rotational viscometer (ARES Rheometric Scientific, strain controlled, couette configuration).

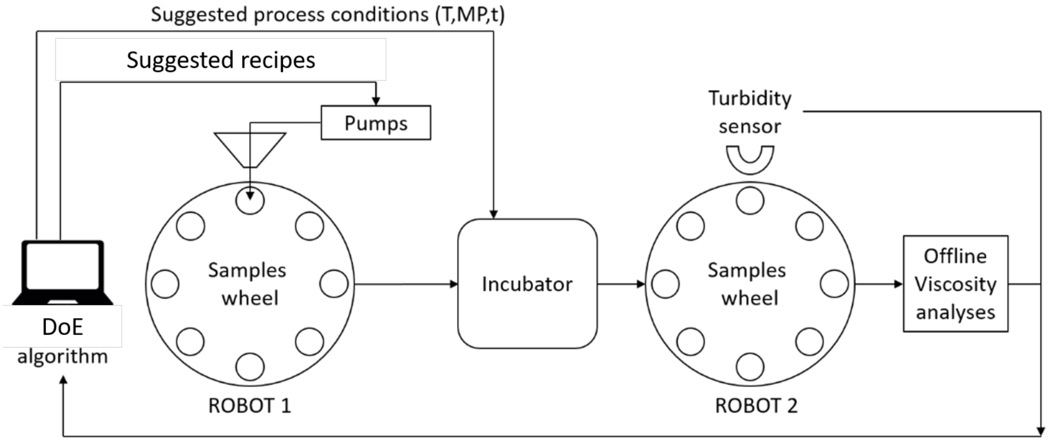


Fig. 3.1 Scheme of the experimental set-up.

3.2.3 Methodology for surfactant selection: Initial Sampling

Initial experiments were performed using a maximin space filling design with the aim of efficiently exploring the entire design space [125]. The constraints of the input variables are given below:

- the sum of the concentrations of the surfactants, of which only three can be non-zero (or active), must be in the range 13 – 15 $g \cdot L^{-1}$.
- P1 concentration must be in the range 0 - 2 $g \cdot L^{-1}$.
- T1 concentration must be in the range 0 - 2 $g \cdot L^{-1}$.

Once the formulated product has been manufactured, it is tested for stability, which has a pass (1) or fail (0) outcome, and viscosity at a shear rate of 10 s^{-1}, which must be between 2,000 and 4,000 $mPa \cdot s$.

3.2.4 Methodology for surfactants selection: DoE algorithm

Objective function

The objective function was inspired by the bridge design reported in Jones et al. [174], which has the dual objectives of optimality with respect to parameter estimation in the model fitted to the response (D-optimality) and space filling. However, the objective function in this work differs in three ways: 1) it does not use the same space filling criteria; 2) it is a weighted objective function; 3) it uses the Bayesian D-optimality objective function [175, 176], which is estimated using Monte Carlo integration [177]. In this work, a bespoke objective function and algorithm has been written to find an optimal design for these experiments, where the objective function consider the two different types of output: a binary response from the stability test and a continuous response from the viscosity test.

For the binary output, there are a variety of models which are suitable to model the discrete response, with the logistic and probit regression being the most commonly used. Initial experimental data were used to identify a suitable model by using forward model selection with the following steps:

1. Fit models to each variable individually and identify the effects with p-values less than a given level of significance (set to 5% in this work).

2. Fit models which contain at least two of the effects identified in step 1, and calculate the AIC, BIC and deviance for these models.

3. Identify the models which minimise the AIC, BIC and deviance.

A logistic regression model with active parameters for the individual effect of S1, S4, P1 and P2 was found to be the best fitting model for this data.

As mentioned in Chapter 2, there are no physical models available for the viscosity test. Analysis of the past experimental data for this test using polynomial regression did not find any suitable models. A Gaussian Process (GP) model was selected as considering the wide range of both linear and non-linear models which could be suitable would be a lengthy process.

Therefore the objective function in this work is given by Eq. (3.1), and has a component relating to a model and a component relating to a distance:

$$\textbf{Ø(D)} = \omega \log(\tilde{E}[U(\mathbf{D})]) + (1-\omega)\log(d(\mathbf{D})) \tag{3.1}$$

where: $\omega \in [0,1]$ is the weight placed on the part of the objective function relating to the stability test response; $(1-\omega)$ is the weight placed on the part of the objective function relating to the viscosity test response; $\mathbf{D}$ is the design scaled to be between 0 and 1; $[U(\mathbf{D})]$ is the estimate of the expected utility for $\mathbf{D}$, which is the part of the objective function related to the stability test responses; and $d(\mathbf{D})$ is the average Euclidean distance between all possible pairs of rows in D, which is the part of the objective function related to the viscosity test responses. The expected utility is given by Eq. (3.3)

$$E[U(\mathbf{D})] = \int u(\theta, y, \mathbf{D})\pi(y \mid \theta, \mathbf{D})\pi(\theta \mid \mathbf{D})d\theta dy \tag{3.2}$$

$$= \int u(\theta, y, \mathbf{D})\pi(\theta \mid \mathbf{D})\pi(\theta \mid \mathbf{D})d\theta dy \tag{3.3}$$

where θ are the parameters in a logit model for the stability test response y, $\pi(y \mid \theta, \mathbf{D})$ is the posterior distribution of the response, $\pi(\theta \mid \mathbf{D})$ is the prior for θ and $\pi(y, \theta \mid \mathbf{D})$ is the joint distribution of y and θ. The utility function, $u(\theta, y, \mathbf{D})$, can be chosen based on the aims of the experiments. In this case, Shannon information gain, which maximises the expected divergence between the posterior and prior distributions, is used as the utility function. The prior for θ is adapted based on the initial experimental results.

Under the assumptions made in this work, $E[U(\mathbf{D})]$ is not analytically tractable, and it is estimated using Monte Carlo integration as shown in Eq. (3.4)

$$\tilde{E}[U(\mathbf{D})] = \frac{1}{B}\sum_{b=1}^{B} u(\theta_b, y_b, D) \tag{3.4}$$

where y_b and θ_b are sampled from $\pi(y, \theta \mid D)$, and B is the number of samples. This estimate is found using *utilityglm* function in the *acebayes* package in R given by Overtall and Woods [177]. Here, we let $B = 1000$.

Space filling designs can be used when a GP model is assumed for the response. Space filling designs impose restriction on the space of, or distance between, points in the design space. In this case, we use the average Euclidean distance as a space filling criteria for the viscosity test response. This distance is calculated using the pairwise distance between rows in the unscaled design, so **D** is converted from the 0 to 1 scaling back to the original scale in the function d(**D**) in Eq. (3.1).

The weight on each of these two components can be adjusted based on the experimenter's aims. For example, if it is assumed that the outcome of the stability test is more important than that of the viscosity test, $\omega < 0.5$ would be appropriated, and vice versa. In this case, we set $\omega = 0.5$ as we treat the two responses equally important.

Point exchange algorithm

Point exchange algorithms find an optimal design by optimising each row of the design with respect to a certain objective function whilst assuming the other rows are fixed [178]. These algorithms perform multiple loops through the design, and continue to optimise rows until stopping criteria are met. In order to avoid any issues with local optima, such algorithms are run for multiple random starting designs. The optimal design is the design found using the algorithm from these random starts which maximises the objective function.

The estimated expected utility, Eq. (3.3), is computationally expensive to calculate, and therefore so is Eq. (3.1), hence we require a computationally efficient method of optimisation. Also, we want to consider samples of possible values of Eq. (3.1) when choosing whether to accept or reject a proposed new row, as Eq. (3.3) is dependent on random samples from the joint distribution of θ and y. We therefore optimise the rows using the Efficient Global Optimisation (EGO) algorithm [179], and accept or reject a proposed row using the Kolmogorov-Smirnov (KS) test [180].

The acquisition function is chosen to balance the objectives of exploring the space where little is known about the function, and exploiting the information we have gained by observing the function at given the points. Bayesian optimisation is demonstrated for a function for a single controllable variable in Fig. 3.2.

The EGO algorithm uses Expected Improvement (EI) [181] as the acquisition function, and continues to add points to the algorithm until the current maximum EI value is less than or equal to 1% of the current maximum objective function value. In this algorithm, we also add a restriction on the number of new points that can be added.

We choose to accept or reject a proposed new row based on a comparison of samples of Eq. (3.1) for a design with and without this new row, which are found by generating R (R=1,000 in this work) samples from the joint distribution of θ and y. The KS test compares two samples to assess whether they come from the same distribution, where the null hypothesis is that these samples come from the same distribution. If the p-value of the KS test is less than α (set to 0.05 in this work), then there is evidence to reject this null at a $100\alpha\%$ significance. Hence, such a p-value for a KS test between two samples of Eq. (3.1) gives evidence to suggest that the objective function distribution after the swap differs from that before the swap, and therefore gives evidence to accept that proposed new row. However, the multiple testing problem will occurs when multiple simultaneous statistical tests are made [182]. The chances of observing a rare event increases, therefore the likelihood of incorrectly

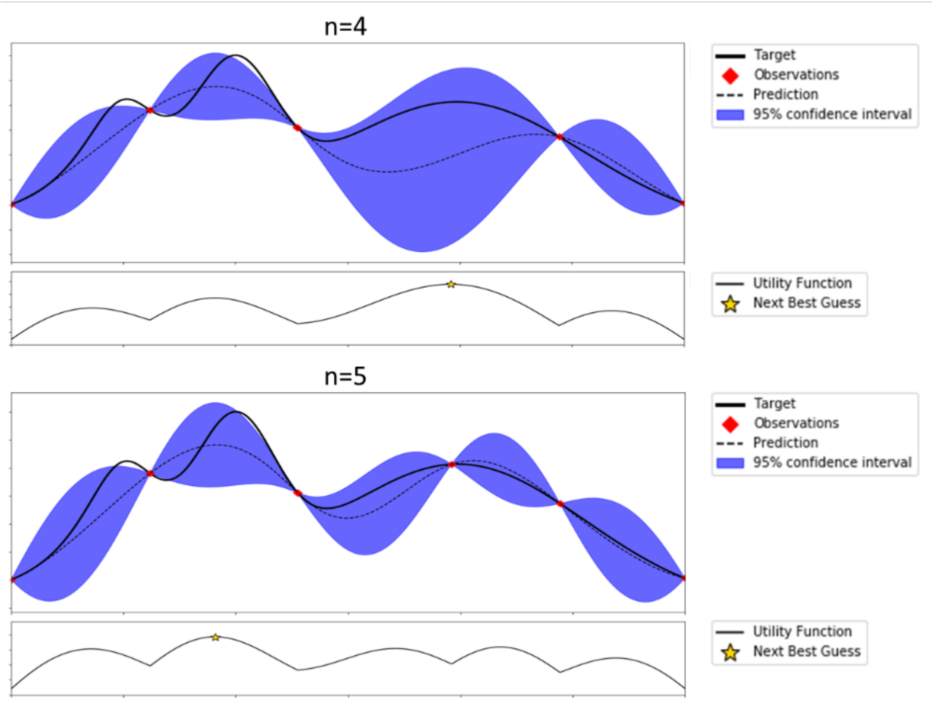


Fig. 3.2 Demonstration of Bayesian optimisation algorithm. The black points are the current evaluations of the function f(x), the solid black line is the Gaussian process estimate of the objective function, the dotted black line is the unknown objective function, and the green line is the acquisition function. The blue shaded area gives the uncertainty in the prediction of the objective function. Note that new points, given by red points, are added where the acquisition function is maximised.

rejecting a null hypothesis increases. Therefore, we also add the condition that the objective function itself must have increased, as we want to find the design which maximises Eq. (3.1). An example of the estimated densities for these samples is given in Fig. 3.3. Therefore,

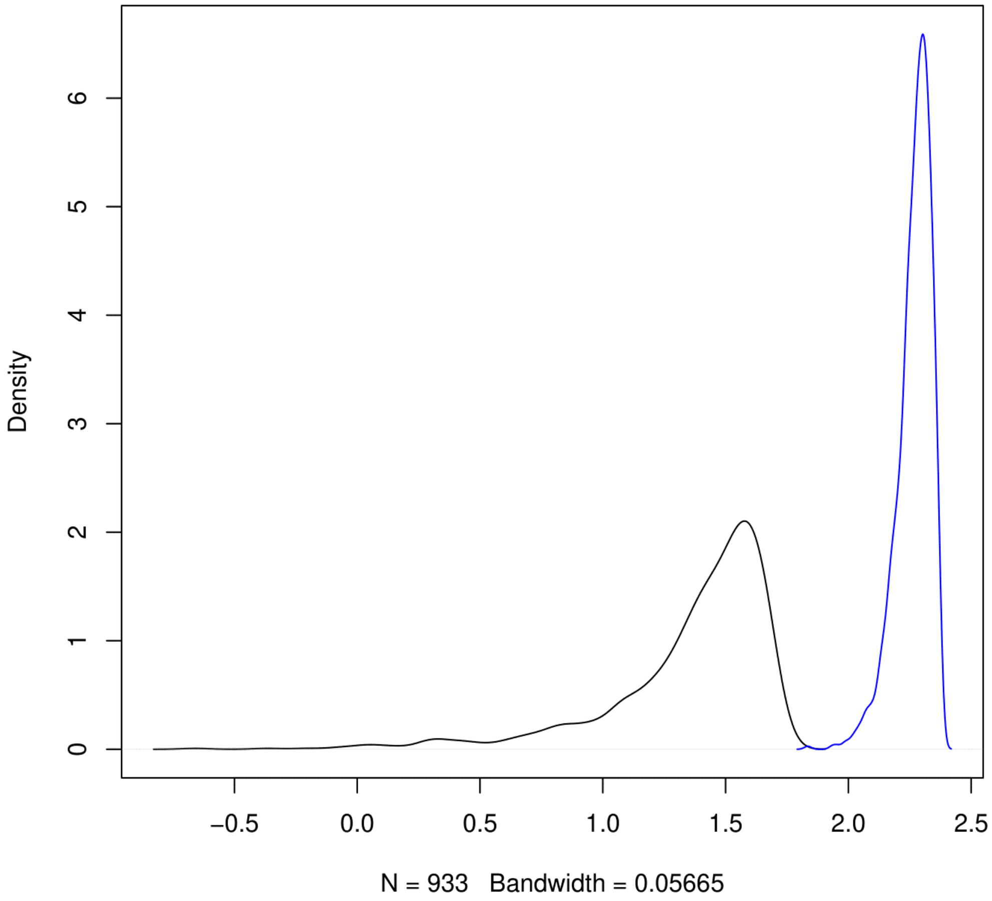


Fig. 3.3 The estimated densities of two objective function samples, one before a swap (black line) and one after a swap (blue line). The KS test for the comparison of these two samples has a p-value of less than 0.05, hence the null hypothesis that these two samples are drawn from the same distribution can be rejected at a 5% level.

the Point Exchange Efficient Global Optimisation (PEEGO) algorithm was proposed and summarized in Scheme 3.1.

```
(1)   Generate a random starting design $\mathbf{D}$.
(2)   For each row, $\mathrm{x_i}$, in $\mathbf{D}$ $\mathrm{i} = 1, \dots, \mathrm{n}$:
  (a)   Let p be a vector of length 10. For each of the 10 possible combinations of three of S1-S5 being active. $\mathrm{j} = 1, \dots, 10$:
    (i)   Generate $\chi_\mathrm{j} = \{x^1, \dots x^T\}$ ($\mathrm{r} = 50$ in this case) random rows to be swapped for $\mathrm{x_j}$, where the jth combination of S1-S5 is active in all elements of $\chi_\mathrm{j}$.
    (ii)  Fit a Gaussian process to $\phi(\chi_\mathrm{j}) = (\phi(\mathbf{D}(\mathrm{x}^1)), \dots, \phi(\mathbf{D}(\mathrm{x}^\mathrm{T})))$ where $\mathbf{D}(\mathrm{x})$ is the design $\mathbf{D}$ with $i$-th row x.
    (iii) Find the row $x_{r+1}$which has the $j$-th combination of S1-D7 active and maximises EI when swapped with $x_i$ (this is done using a constrained optimisation algorithm in the package *DiceOptim* in R)
    (iv)  If the maximum EI is less then 0.01 $\max(\phi(\chi_j))$ and $its_{EGO} \geq maxits_{EGO}$ ($maxits_{EGO} = 100$ in this work), do not amend $\chi_j$and repeat from step (ii)
    (v)   Let $x_j^*$ be the element of $\chi_j$ for which $\phi(\chi_j)$ is maximised. Add this to a set $\chi^*$.
    (vi)  Let the $j$-th element of $\boldsymbol{p}$, $p_j$, be the p-value of the KS test for the comparison of samples of $\phi(\boldsymbol{D}(x_j^*))$.
  (a)   Find the $x^* \in \chi^*$ which relates to $p^* = \min(\boldsymbol{p})$.
  (b)   If $p^* < \alpha$ ($\alpha = 0.05$ in this work) and $\phi(\boldsymbol{D}(x^*)) > \phi(\boldsymbol{D}(x_i))$, then swap $x_i$ for $x^*$. Otherwise, do not change $x_i$.
3.    Let $D^*$ be the design after all rows have been optimised using step 2.
4.    If:
  •     The p-value of the KS test for comparison of the samples of $\phi(\boldsymbol{D})$ and $\phi(\boldsymbol{D}^*)$ is less than $\alpha$,
  •     $\phi(\boldsymbol{D}^*) > \phi(\boldsymbol{D})$,
  •     The average absolute difference between all pairwise values in $\boldsymbol{D}$ and $\boldsymbol{D}^*$ is less than a tolerance $t$ ($t = 0.01$ in this work), and
  •     $its_{PE} \geq maxits_{PE}$ ($maxits_{PE} = 100$ in this work),
      Then stop the algorithm and return $\boldsymbol{D}^*$ as the optimal design. Otherwise, let $\boldsymbol{D}^*$ be $\boldsymbol{D}$ and $its_{PE} = its_{PE} + 1$ and return to step 2.
```

Scheme 3.1 Scheme of Point Exchange algorithm.

3.2.5 Product and process design: Initial sampling

A batch of 224 experimental data at fixed process conditions (T = 50 $°C$; MP = 300 rpm; t = 2 h) was used to train the algorithm and start the iterative process. This data set was already available from Chapter 2. The previous explorative experimental campaign devoted to optimise the process without taking into account the influence of the process conditions. The choice of using these previously collected data is justified to mime a common situation in products development, in which first preliminary investigations provide an initial data set. Other 12 training data were collected at a fixed recipe (S1 = 5.62 g L^{-1}; S2 = 2.50 g L^{-1}; S3 = 6.88 g L^{-1}; P1 = 0.90 g L^{-1}; T = 2.00 g L^{-1}) at different process conditions. Hence, the overall training data set consisted of 236 data points.

3.2.6 Product and process design: DoE algorithm

Thompson-sampling efficient multi-objective (TS-EMO) optimization algorithm was chosen as the design of experiments algorithm. A detailed presentation of the algorithm can be found in Chapter 2 [146]. Briefly, the iterative algorithm consists of the following steps:

- Train Gaussian processes for each of the outputs to be optimized based on an initial dataset.
- Sample functions from the obtained GPs using Thompson spectral sampling.
- Find the Pareto front of the sampled functions.
- Find the points that are predicted to give the largest improvement of the hypervolume.
- Test experimentally the selected data points and add them to the training set.

The optimization procedure can be stopped when the maximum number of evaluations is reached or when the operator is satisfied with the obtained results. This can be automated by terminating the algorithm when the objective functions are lower than a given epsilon. For this specific case study, the number of suggested experiments at each iteration was set equal to 8. The input variables were chosen as the concentrations of the ingredients and the three targets to minimise were chosen as the turbidity value in NTU, the squared distance between the measured viscosity and the target viscosity of 3 $Pa \cdot s$, and the cost of the adopted ingredients ($\$ \cdot L^{-1}$). The latter was calculated as the sum as the unitary prices of the ingredients multiplied by the adopted amounts in each sample. As this target was an explicit function of the input variables, the code was modified to not train a GP for this specific target, using the directly calculated values instead. The targets were chosen according to the indications of the product supplier: an ideal product is a stable homogeneous clear formulation with a viscosity as close as 3 $Pa \cdot s$, at the lowest possible cost.

3.3 Results and discussion

3.3.1 Surfactant selection

As explained in Section 3.2.4, the algorithm was trained using a maximin space filling design consisting of 230 data points. 52% of the samples in the training data set were stable, 12.61% met the viscosity target, and only 3.48% passed both criteria. The algorithm was run in order to suggest an experimental bridge design of 48 samples. In Fig. 3.3, it is shown that the blue line (the new design) in the plot shifted to the right of the black line (the initial design). In conjunction with the p value for the KS test, it suggested that by applying the point exchange algorithm, the new design can give more information about the system than the original design.

In Table 3.1, we show a comparison between the percentage of samples passing stability and viscosity criteria in the training data set and in the suggested DoE. In the latter, 91.67%

of the samples were stable, and 12.5% passed both criteria. This may ascribed to the fact that, in the initial design, we select experimental points according to a space-filling criterium, without taking into account the information gain of the models. Therefore, in order to gain more information of the system, the algorithm seeks to explore the part never seen in the initial dataset, more likely to consist of stable samples in the right viscosity range.

Table 3.1 Comparison between the training data set and the suggested DoE.

	Maximin DoE	Bridge-Design DoE
Stability test: passed Viscosity test: passed	3.48%	12.50%
Stability test: failed Viscosity test: passed	9.13%	0.00%
Stability test: passed Viscosity test: failed	48.7%	79.17%
Stability test: failed Viscosity test: failed	38.69%	8.33%

The resulting data-set was used to train a Gaussian Process model for the prediction of viscosity of samples and to guide the optimization of the formulation, within three experimental iterations. The GP model was chosen is due to its flexible nature in modelling complex system as well as its good performance in modeling formulation viscosity property as shown in Chapter 2. To mimic a common situation in product development, some prior knowledge was included in the data set. In fact, it was known that mixtures of the surfactants without any added polymer and thickener show a water-like viscosity. The trained GP was used to predict the viscosity response over the entire input variable space. The training procedure was similar to the one detailed explained in Chapter 2, Section 2.2.6. The stationary kernel functions from the Matérn class was chosen as the kernel function in this case as well.

At each iteration the candidates predicted to be closer to the midpoint of the desired target range (2.0 – 4.0 $Pa \cdot s$) were selected. Using the trained classifier, solutions predicted to be unstable were discarded, and the resulting 20 best experiments were tested experimentally. In total, three iterations were run and 60 experiments were carried out.

In Table Table 3.2 the formulations passing both criteria are reported. All 60 experiments resulted in clear, stable formulations, 20% of which passed the viscosity test.

Table 3.2 Formulation passed both criteria in the three DoE iterations.

S1	S2	S3	S4	S5	P1	T1	Viscosity	Iteration
$\mathrm{g\,L^{-1}}$	$\mathrm{g\,L^{-1}}$	$\mathrm{g\,L^{-1}}$	$\mathrm{g\,L^{-1}}$	$\mathrm{g\,L^{-1}}$	$\mathrm{g\,L^{-1}}$	$\mathrm{g\,L^{-1}}$	$Pa \cdot s$	
1.66	0.00	8.34	0.00	5.00	2.00	1.30	3.60	1
1.66	0.00	0.00	6.66	6.66	1.80	1.60	2.52	1
5.00	0.00	0.00	5.00	5.00	2.00	0.90	2.38	1
5.35	0.00	6.43	3.21	0.00	1.29	1.71	2.86	2
0.00	5.36	4.29	0.00	4.29	0.29	1.57	2.75	2
0.00	3.21	0.00	6.43	5.36	1.43	1.86	3.36	3
2.14	6.43	0.00	0.00	5.36	0.00	1.86	3.13	3
0.00	6.43	0.00	3.21	5.36	2.00	1.86	2.72	3
7.50	0.00	1.07	5.36	0.00	1.00	1.71	2.69	3
2.14	0.00	7.50	0.00	4.29	1.57	1.29	3.62	3
5.36	3.21	6.43	0.00	0.00	0.00	1.71	3.28	3
10.71	0.00	0.00	0.00	4.29	0.00	1.57	2.49	3

At this point, it is worth stressing that a good number of candidates was obtained within a total number of 338 experiments carried out in 17 working days, without any need for a physical model for properties prediction. Here we can use the case study in Chapter 2 as a reference, since a similar system was used as the case study. In Chapter 2, the recipe of the formulation was optimized to obtain a stable, clear, formulation with the same target viscosity, but without allowing for a free choice of the surfactants. Only three surfactants were available corresponding to S2, S3, and S5 used in this work. The optimization procedure was started using 96 LHC experiments and 128 further experiments collected in 16 iterations of the optimization algorithm, with a total of 224 experiments. Although in the reported paper the formulation was also optimized with respect to its price, it is worth noticing that in the current work the choice of three surfactants from five available makes the input space variable one order of magnitude larger. However, thanks to the adoption of a maximin design coupled with a bridge-design approach the total number of required experiments only increased to 338. The examination of the solutions also gives an insight into the role of the different ingredients in the processed formulations. The lowest occurrence of surfactants S2, S3, and S4 suggests that interactions of these compounds with the other components have a higher probability to form unstable, turbid mixtures. As expected, the thickener T1 is responsible for higher viscosity of the samples and its concentration tends to be close to the upper limit of the adopted constraints; however, contrary to suggestions of human experts, the algorithm was able to find good solutions also using a concentration of T1 lower than 2 $g\,L^{-1}$, which significantly decreases price of the final product. Interestingly, although polymer P1 was considered by human experts to be responsible for the increase in viscosity, when certain combinations of surfactants are adopted, the polymer concentration can be reduced, suggesting that interactions between these ingredients are contributing to the increase in viscosity. As shown, this preliminary analysis gave some qualitative insight about the physics of the system.

3.3.2 Product and process design

15 iterations of the closed-loop optimization (120 samples) were carried out in order to assess the predictive performances of the obtained surrogate models and to evaluate the results of the optimization. Among the conditions suggested by the algorithm, 7 non-dominated

solutions were found (Table 3.3) in the experimental Pareto front of the data set.

Table 3.3 Non-dominated experimental data suggested by the TS-EMO algorithm.

S1	S2	S3	P1	T1	T	MP	t	Turbidity	Viscosity	Price
g L^{-1}	g L^{-1}	g L^{-1}	g L^{-1}	g L^{-1}	°C	rpm	min	NTU	mPa·s	\$ L^{-1}
2.8	3.2	9	0.4	1.46	35	137	77	152	2,409	1.81
1.09	2.45	11.45	1.2	1.92	54	211	118	157	3,424	1.79
0	0.52	14.48	0.4	1.33	30	220	42	28	197	1.51
0	2.01	12.99	0.01	1.24	51	180	73	31	409	1.45
5.1	5.89	4.01	0	1.97	49	230	70	24	3,681	2.05
5.1	5.65	4.25	0.76	1.91	49	230	83	25	3,071	2.14
0.93	5.63	8.44	0.94	1.94	52	206	115	30	2,337	1.74

All of them were clear, homogeneous samples, with no phase separation, which is one of the requirement for the commercial products. five out of the 7 non dominated solutions satisfy the viscosity requirement to be in the range 2,000 – 4,000 mPa·s. The presence of two samples with a viscosity lower than 1,000 mPa·s is justified by their relatively low price and turbidity, which is in agreement with the interpretation of the Pareto front as a trade-off between different conflicting objectives. Further criteria can be used to discriminate between the obtained best solutions. It is also worth noticing that all the proposed solutions were obtained using a similar or lower temperature with respect to the previous data included in the training set and, most importantly, with a significant reduction of the needed time and mixing power. The non-dominated solutions that passed the viscosity and turbidity criteria in the training set are reported in Table 3.4.

As one can see comparing Table 3.3 and Table 3.4, among the solutions that passed both criteria, the ones suggested during the optimization procedure have a lower average price of the ingredients, which together with the advantages in terms of needed mixing power and processing time, might have a significant effect on the overall economics and productivity.

Table 3.4 Non-dominated experimental data in the training set.

S1	S2	S3	P1	T1	T	MP	t	Turbidity	Viscosity	Price
g L^{-1}	g L^{-1}	g L^{-1}	g L^{-1}	g L^{-1}	°C	rpm	min	NTU	mPa·s	\$ L^{-}1
3.97	3.16	7.87	0.84	1.6	50	300	120	11	2,678	2
4.61	2.62	7.76	1.24	1.04	50	300	120	28	3,574	2.06
8.4	3.18	3.43	0.48	1.8	50	300	120	18	2,948	2.43
2.79	2.93	9.28	1.72	0.88	50	300	120	26	2,632	1.92
10.94	3.94	0.12	1.8	1.36	50	300	120	28	2,992	2.8
4.02	3.09	7.89	0.99	1.34	50	300	120	15	2,824	2
3.01	0.64	11.35	0.53	1.75	50	300	120	50	2,789	1.87
3.06	1.96	9.95	0.31	1.42	50	300	120	23	3302	1.82

The predictive performances of the adopted models were evaluated comparing the predicted Pareto front of the GPs with the experimental non-dominated solutions, Fig. 3.4 and Fig. 3.5.

The satisfactory predictive performances can enable further improvements in formulations design, based on different a posteriori criteria that may be selected by human intuition and/or expertise.

3.4 Conclusions

In this work, we present algorithm which can be seen as extensions on the previous work in Chapter 2 in order to solve two critical problems in formulated product design:

1. Can we use an algorithmic design of experiments (DoE) approach, coupling statistical models with robotic experiments, to guide emulsion design and efficiently select ingredients?

2. Is it possible to train a surrogate statistical model to describe the desired product

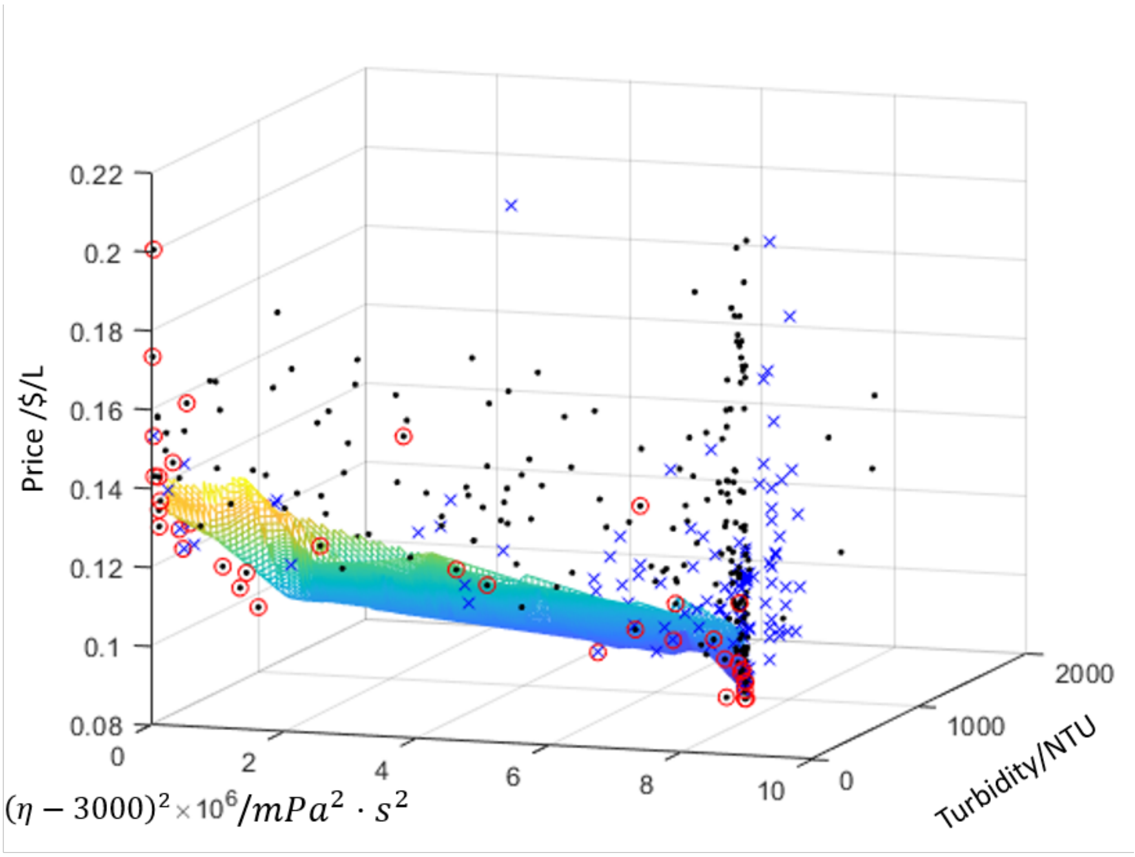


Fig. 3.4 3D visualization of the predicted Pareto front of the GPs with the experimental non-dominated solutions.

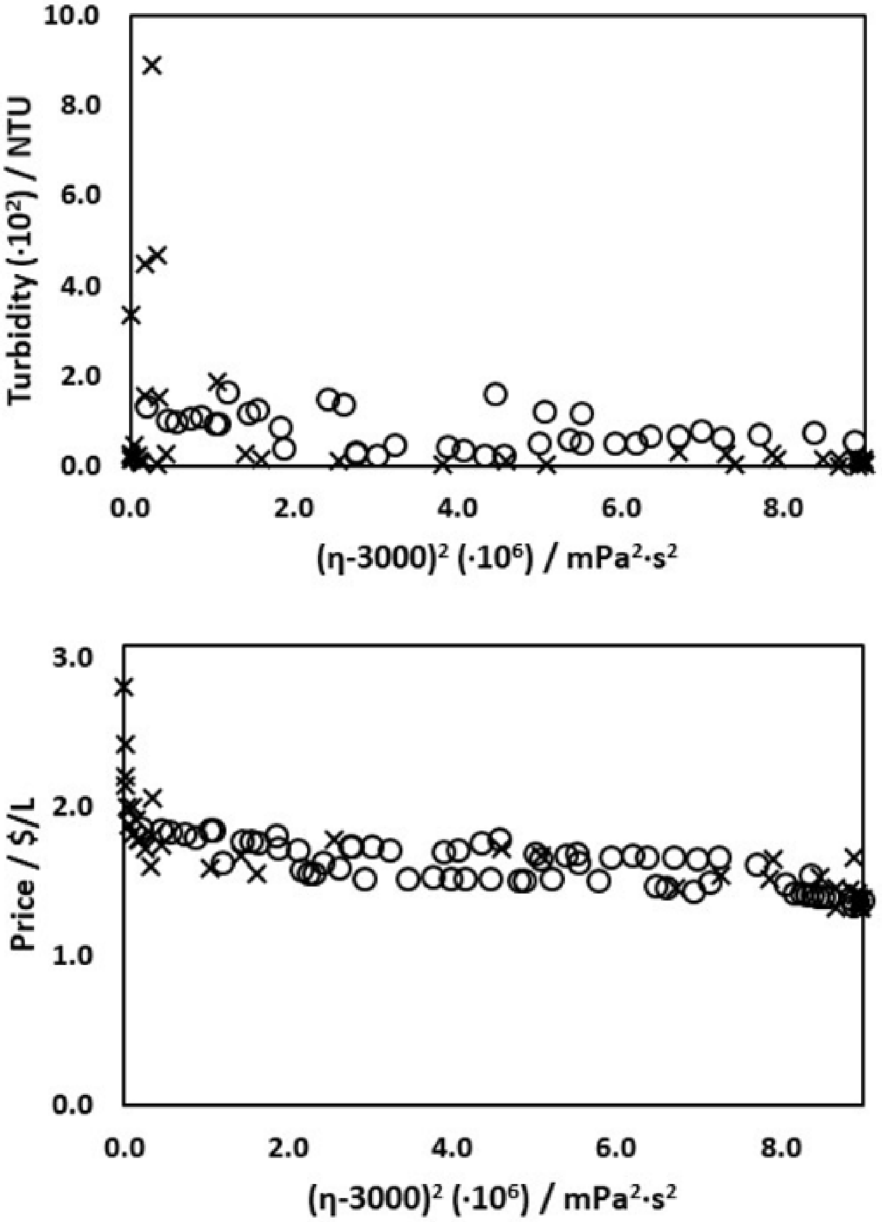


Fig. 3.5 Two projections of the experimental (X) and predicted (O) Pareto front.The x-axis was chosen to be viscosity was due to the fact that this setting can give best visualisation of the Pareto front.

properties in the absence of a physical model, and use it to optimize the product and process design in an automated fashion?

The proposed Point Exchange Efficient Global Optimisation (PEEGO) algorithm was used to find a Bridge-design of experiments to maximize the information gain, in order to find suitable solutions for a commercial formulated product. It was tested with the design of the same commercial liquid formulated product in Chapter 2, in this case, only three surfactants can be chosen from an ingredients library. A logistic model and a Gaussian process model was selected to describe a discrete and a continuous target of the product, i.e. stability and viscosity. The PEEGO algorithm was then applied to simultaneously optimise the information gain for the two responses.

A cheap-to-evaluate GP was trained using the experimental results, and used to predict the viscosity response over the entire input variable space. This triggered an iterative process that allowed to increase the percentage of samples passing both quality criteria from 3.68% (maximin DoE) and 12.50% (bridge-design DoE), to 20.00 % over 60 samples obtained in 3 iterations. This outperformed the results previously obtained for a similar case study, using a Latin Hypercube sampling approach coupled with an iterative procedure, in the absence of a bridge-design approach.

In addition to the good number of candidates obtained in a short time in the absence of physical predictive models, the a posteriori analysis of the obtained solutions gives some qualitative physical insight to the role of the different ingredients and their non-trivial complex interactions. Further research will be needed to rationalize this information using systematic approaches for the generation of physical knowledge from fast automated development of formulated products.

In order to tackle the co-optimization of both the formulation recipe and the process conditions simultaneously, TSEMO algorithm was applied for optimization. The proposed methodology enabled to identify new formulations meeting the discrimination criteria. Moreover, the suggested samples were generally cheaper and required less mixing power and process time, which can generate considerable profits on scale. The overall optimization procedure was completed within two weeks, avoiding human bias and using automated operations, exploring a complex, multidimensinal chemical space. As a result, the developed methodology can lay the ground work for a faster and more efficient process development and a consequent early product release time, without requiring any extensive expertise of the human operators. GPs were confirmed to be suitable surrogote model to predict the complex relationships between the input variables, and the target properties, when no physical models were available.

In Chapter 2 and Chapter 3, the presented work is stressing the need of using the results of black-box optimization and robotic experimental campaigns to optimise the input variables in order to achieve desired product properties. In the following Chapter 4 and Chapter 5, we will discuss the potential methods for understanding physical knowledge of the investigated systems, and automated identification of physical laws from raw experimental datasets.

Chapter 4

Identifying Key Components in Complex Systems using Feature Engineering Method

4.1 Introduction

The nature of chemical product design problems is diverse and multidisciplinary. The aim of chemical product design is to find a product that exhibits certain specified behaviour/properties, corresponding to the desired functional properties [183]. Thus, in the area of formulated consumer products the main useful functions, for example the function of "UV protection", are achieved through the use of molecules and particles with corresponding physico-chemical properties, e.g., UV absorbance and scattering of TiO_2 micro-particles reducing the flux of the harmful range of UV solar radiation to the skin [184].

Several important technical challenges in the design of formulated products stem from the fuzziness of performance criteria for a number of desired functions, not easily converted to numerical specifications, and their apparent complexity, which does not allow easy prediction of properties based on composition [185]. The latter means that the performance property is often described as a "system property": a property that emerges due to interactions among individual components of a system i.e., the ingredients of a formulated product in our case [24].

The UV protection of a sun cream product is jointly influenced by various features originating from the composition of the product, the rheology performance and etc. There are two categories of UV radiation: UV-A and UV-B, which constitutes around 10% of the total solar radiation [186]. Overexposure to UV radiation can lead to harmful clinical consequences which include photo-ageing [187], immune suppression [188], skin cancers [189] and etc. To avoid this, sunscreen products play a crucial role in reducing the risk of melanoma and photo-ageing by absorbing UV rays [190]. Sun protection factor (SPF) and protection grade of UVA (PA) are the two main indicators for describing the sun cream products' ability to protect against UVA and UVB. It is crucial to understand the key factors in the complex system which highly influence the two main indicators. And this will lead to the development of new sun protection formulation to obtain the required SPF and PA factor rapidly and inexpensively . Due to the fact that experimental testing is time-consuming and expensive, multiple models were proposed to calculate the SPF and PA values. Herzog and Ostervalder proposed a method to calculate the SPF and PA values using the concentration of the UV filter substances based on physical model [191].By applying the physcial model, the mean squared error of the SPF prediction for the oil in water (O/W) type sunscreen product was 9.47 However, due to the complex nature of the sun cream products, various external factors are also know to influence SPF and PA values, such as type of product [184], amount of emollient and solvent [192], and etc. was neglected in the model. Shim et al. introduced machine learning method in SPF and PA prediction [193]. A decision tree regression algorithm was applied for the prediction of SPF and PA values. The mean squared error of the prediction of SPA and PA values on the test dataset was 60.2 and 48.5 respectively. The application of ML techniques increased over the last two decades due to various factors, e.g. the availability of large amounts of complex data with little transparency [194] and the increased usability and power of available ML tools [195]. It shows good performance in navigating high dimensionality space even without *a-priori* knowledge. Moreover, by applying feature engineering techniques, it can aid in deriving pattern from existing dataset [196], which can also provide a basis for development approximations in future formulation product development.

In this work, by applying advance machine learning techniques, it is expected that some basic insights about the dataset could be gathered from the feature engineering results so that better understanding about the sun cream products UV-protection natures can be achieved.

4.2 Methodology

4.2.1 Feature Engineering: Principle and Methods

Feature Visualisation: Exploratory Data Analysis

Feature visualisation is a helpful technique that can provide a comprehensive understanding of the feature. However for a high dimensional feature space, visualising the feature space clearly is not an easy task. So instead of visualizing the feature space at one time, it is recommended to analyse the pairwise correlation through exploratory data analytics (EDA), where calculate correlation matrix is one of the most common techniques [197]. The correlation matrix is a square matrix based on Pearson product-moment correlation coefficients (Pearson correlation coefficient, for short), which is a measure of the strength of a linear association between two variables and is denoted by r [198]. The Pearson correlation coefficient, is calculated with the following formula:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{4.1}$$

Pearson's r value can take a range of values from +1 to -1. A value of 0 indicates that there is no association between the two variables: x, y. A value greater than 0 indicates a positive association; that is, as the value of one variable increases, so does the value of the other variable. A value less than 0 indicates a negative association; that is, as the value of one variable increases, the value of the other variable decreases. Based on such interpretation, Pearson's r based EDA can help to get some basic insights about the linear correlations between outputs and features; hence, the features that are relatively high related to outputs can be chosen as "exploratory feature" for further ML model construction.

Feature Selection: Random Forest

In the case where the given datasets are high dimensional, it is always suggested to conduct dimensionality reduction through feature selection. The basic idea of feature selection is to remove the features that have less influence on the performance of ML models while only keep the features that are most influential on the ML models. Again it have to be underlined that when ML models are different, the selected features are usually different as well. Random forest method is one of the most powerful feature selection techniques, which construct with individual decision tree models at training stage. In this report, only random forest algorithm is used as an example to show how feature selection works. The schematic random forest algorithm is shown in Fig. 4.1, it can be seen that random forest is essentially an ensemble model of decision tree models. The feature importance is calculated as the decrease in node impurity weighted by the probability of reaching that node. For each decision tree within random forest model, the node importance is calculated using Gani

Importance which is shown below:

$$NI_j = w_j C_j - w_{left(j)} C_{left(j)} - w_{right(j)} C_{right(j)} \tag{4.2}$$

where NI_j is the importance of node j, w_j is the weighted number of samples reaching node j, C_j is the impurity value of node j, $left(j)$ is the child node from right split on node j, and $right(j)$ is the child node from right split on node j.

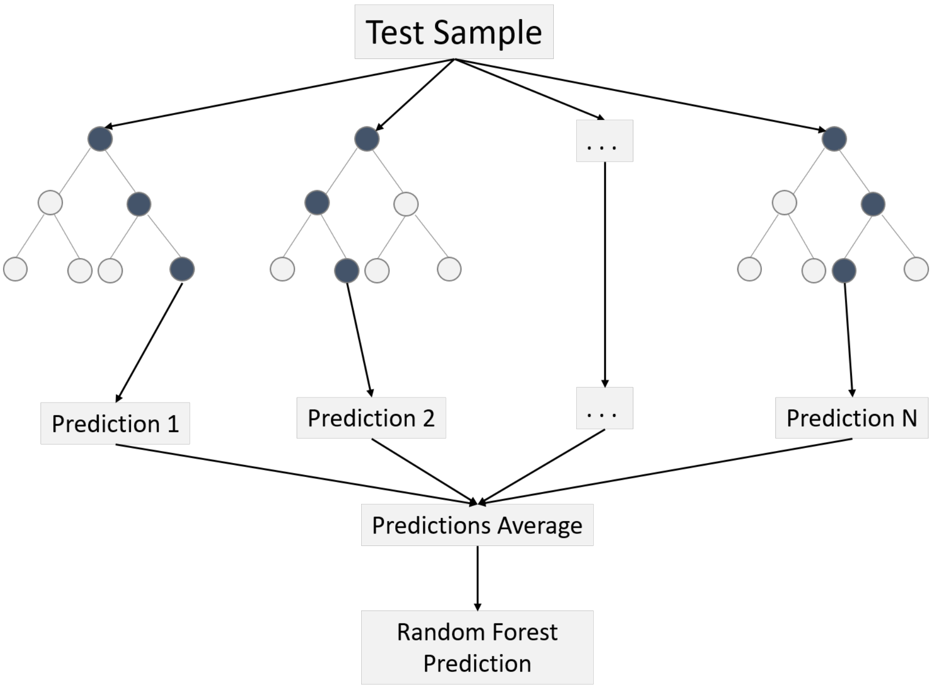


Fig. 4.1 Schematic structure of random forest algorithm. Adopted from Afroz Chakure's blog (Available at https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f. Last accessed: 20th Mar 2021).

The importance for each feature on a decision tree is then calculated as

$$FI_i = \frac{\sum_{j:nodejsplitsonfeaturei} NI_j}{\sum_{k \in allnodesNI_k}} \tag{4.3}$$

where FI_i is the feature importance of feature i, NI_j is the importance of node j.

These can then be normalized to a value between 0 and 1 by dividing by the sum of all feature importance values:

$$normFI_i = \frac{FI_i}{\sum_{j \in allfeatures} fi_j} \tag{4.4}$$

Therefore the final feature importance for the random forest model is the average over all the decision trees. The expression is indicated as below:

$$RF_i = \frac{\sum_{j \in alltrees} normFI_{ij}}{T} \tag{4.5}$$

where RF_i is the importance of feature i calculated from all decision tress in the RF model, $normFI_i j$ is the normalized feature importance for feature i in tree j, and T is the total number of the decision trees. The detailed explanation of random forest can be found in Ref [199].

Feature Extraction: PCA

Besides feature selection, feature extraction is another useful tool for dimensionality reduction in feature engineering area. Compared to the former two methods, feature extraction aims to create a new feature space by projecting the original feature space with certain rules. Principal Component Analysis (PCA) is a powerful tool, and therefore is chosen as an illustrative example. Implementation of other feature extraction techniques, such as linear discriminate analysis and kernel principal component analysis, can be done in a similar manner [200].

As its name implies, PCA aims to find the principal component of the features in the sense that the covariance between such component and outputs are largest [201]. In PCA, a D×K dimensional transformation matrix W is constructed to convert the original feature space $x = [x^1, x^2, \cdots, x^D]$ into a new feature space $\hat{x} = [x^1, x^2, \cdots, x^K]$to facilitate further analysis. Usually, the transformation matrix is constructed based on the covariance matrix between different features. The covariance between features x^i and x^j can be calculated as:

$$\sigma_{ij} = \frac{1}{n}\sum_{k=1}^{n}(x_k^i - \overline{x^i})(x_k^j - \overline{x^j}) \tag{4.6}$$

Based on such covariance definition, a D×D dimensional covariance matrix from feature space $x = [x^1, x^2, \cdots, x^D]$ can be obtained then by choosing the K largest eigenvalues and the corresponding eigenvectors, the transformation matrix could be constructed. In such a framework, the feature importance is defined as the ratio between its corresponding eigenvalue and the overall sum of all eigenvalues.

$$\frac{\lambda_i}{\sum_{i=1}^{D}\lambda_i} \tag{4.7}$$

4.2.2 Model Fitting: Method and Hyper-parameter tuning

Six types of model were applied for SPF and UVA prediction, namely ridge regression, Bayesian regression, supporting vector machine (SVM), k-nearest neighbours (k-NN) regression, decision trees (DT), and Gaussian process regression (GP). These six types of models were chosen was due to their diversity in fitting strategy.

Ridge regression is a type of linear models in which the target value is expected to be a linear combination of features [202],

$$\hat{y}(\omega, x) = \omega_0 + \omega_1 x_1 + \cdots \omega_n x_n \tag{4.8}$$

In the ridge regressor, the ridge coefficients minimize a penalized residual sum of squares:

$$\min_{w} ||X\omega - y||_2^2 + \alpha||\omega||_2^2 \tag{4.9}$$

where ω is the coefficient parameters,$\alpha \geq 0$ is a shrinkage coefficient, the larger α it is, the greater the amount of shrinkage and thus the coefficients become more robust to collinearity. α was tested from 10^{-3} to 10^{3}, and an optimal value was selected for the model fitting.

Bayesian regressor estimates a probabilistic model of the regression problem [203]. The prior for the coefficient ω is given by a spherical Gaussian distribution:

$$p(\omega|\lambda) = N(\omega|0, \lambda^{-1} I_p) \tag{4.10}$$

the priors over α and λ are chosen to be gamma distributions, the conjugate prior for precision of the Gaussian distribution.

SVM regression performs linear regression in the high-dimension feature space using ε-insensitive loss, and tries to reduce model complexity by minimizing $||\omega||^2$ [204]. This can be described by introducing (non-negative) slack variables $xi_i\ xi_i^*, i = 1, \cdots, n$, to measure the deviation of training samples outside ε-insensitive zone. Thus, the SVM regression is formulated as minimization of the following function:

$$\min \frac{1}{2} ||\omega|| + C \sum_{i=1}^{n} (\xi_i + \xi_i^*) \tag{4.11}$$

$$s.t. y_i - f(x_i, \omega \leq \varepsilon + \xi_i^*) \tag{4.12}$$

$$f(x_i, \omega) - y_i \leq \varepsilon + \xi_i \tag{4.13}$$

$$\xi_i, \xi_l^* \geq 0, i = i, \cdots, n \tag{4.14}$$

This optimization problem can be transformed into the dual problem and its solution is given by

$$f(x) = \sum_{i=1}^{n^s p} (\alpha_i - \alpha_i^*) K(x_i, x) \tag{4.15}$$

$$s.t. 0 \leq \alpha_i \leq C, 0 \leq \alpha_i^* \leq C \tag{4.16}$$

where $n_S V$ is the number of support vectors (SVs), and K is the kernel function.

$$K(x, x_i) = \sum_{j=1}^{m} g_j(x) g_j(x_i) \tag{4.17}$$

Three different types of kernels were tested, namely linear, RBF, polynomial kernel. For the RBF kernel, parameter γ and C were tested within range of $[10^{-3}, 10^3]$, and for the polynomial kernel, another parameter *degree* was varied between 1 and 10.

KNN regressor uses feature similarity to predict values of the new data points, which means that the new point is assigned a value based on how closely it resembles the points in the training dataset [205]. Euclidean distance was selected for the distance calculation. The key parameter which define the number of n-neighbours was test within range of 1 to 100.

A decision tree is a type of supervised machine learning model, which is used to predict a target by learning decision rules from features. A decision tree is constructed by recursive partitioning, which means that the tree will split from the root node, and each node will be split into left and right child node [206]. A maximal depth of the tree will be set as a limit when a decision tree is pruned. In order to split the nodes at the most informative features, the objective function is to maximize the information gain at each split, which is defined as below:

$$IG(D_p, f) = I(D_p) - \left(\frac{N_{left}}{N_p} I(D_{left}) + \frac{N_{right}}{N_p} I(D_{right}) \right) \tag{4.18}$$

where f is the feature to perform the split, D_P, D_{left} and D_{right} are the datasets of the parent and child nodes, I is the impurity measure, N_p is the total number of samples at the parent node, and N_{left} and N_{right} are the number of samples in the child nodes. For Decision Tree model, different values of n between five and 15, where n is the number such that impure nodes must have n or more observations to be split, were tested at a tree fitting stage. The best level of pruning was estimated through cross validation (where the best level is the one that produced the smallest tree that is within one standard error of the minimum-cost subtree).

Gaussian process regressor is a nonparametric kernel-based probabilistic model which is based on Gaussian processes (GP). The prior of the GP need be specified based on the training dataset, and the prior's covariance is specified by passing a kernel object. The hyperparameters of the kernel then are optimized based on maximizing the log-marginal-likelihood, given by

$$\log p(y|x,\theta) = -\frac{1}{2}y^T K^{-1} y - \frac{1}{2}\log|K| - \frac{n}{2}\log 2\pi \tag{4.19}$$

where K is the covariance matrix, θ is the vector of hyperparameter, n is the number of data points [160].

4.3 Experimental Dataset

The dataset was provided by BASF SE. The data is a set of mixtures with two continuous system properties of the resulting product, namely SPF *in vivo* and UVA-PF *in vitro*. Based on the function of the ingredients, the dataset was sorted based on the concentration of each function. We start the analysis from O/W type sun cream products only, which contains 131 formulations. Each of the data can be described as:

- A combination of proportions of five functions of the ingredients (contagious variables): high polarity emollient, medium polarity emollient, low polarity emollient, UVA filter, UVB filter.

- Emollient concentration: Emollients were categorised into three groups by its polarity: high polarity, medium polarity and low polarity. The concentration of the emollients in the same group was summed.
- UVA and UVB filters were represented by the overall absorbance of a UV filter composition via Beer-Lambert's law.

  $$A = \varepsilon l c$$

  where ε is the molar attenuation coeffiecnt of absorptivity of the attenuating species, l is the optical path length, c is the concentration of the attenuating species.
- UV filter ratios (continuous variables): UVB filter vs UVA filter (+broad spectrum filter), UV filter in oil phase vs UV filter in water phase and absorbing type UV filter vs scattering/reflecting type UV filter were included.
- A description of the UV protection of the final product (continuous variables): SPF *in vivo* and UVA-PF *in vitro*.

An illustrative dataset for the O/W type sunscreen products is summarized in Table 4.1.

Table 4.1 Summary of O/W type sunscreen product dataset.

	Outputs		Inputs								
	SPF *in vivo*	UVA-PF *in vitro*	Visocosity (mPa*s)	Emollient high	Emollient low	Emollient medium	Scattering /Absorber in UV filters	UV filter in water	UVA/ UVB	UVB filter	UVA filter
count	131	131	131	131	131	131	131	131	131	131	131
mean	43.78	17.59	18447.56	21.40	12.69	2.81	0.28	0.37	0.90	6.46	5.13
std	16.91	6.73	18221.39	87.45	90.70	4.63	0.28	0.24	0.69	4.37	3.16
min	9.00	6.20	200.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	31.00	11.50	5750.00	8.00	2.00	0.00	0.00	0.22	0.50	3.00	3.50
50%	45.00	17.70	12000.00	14.00	4.00	0.00	0.28	0.38	0.83	6.00	5.00
75%	57.00	22.30	26000.00	19.00	5.00	4.00	0.46	0.50	1.25	10.00	8.00
max	90.00	38.60	105000.00	1010.00	1042.00	19.00	1.00	0.84	3.50	21.00	10.00

We further analyse the gel type sun cream products to show that the method can be applied to other systems with minor changes. This dataset contains 88 data points, and each of them can be described same as above.

Then we move to a larger dataset which consisted of different formulation types of the sun cream product. The overall dataset included 261 formulations and each formulation was described as:

- Formulation type (categorical variables): O/W type, W/O type, gel, alcohol and anhydrous.
- A combination of proportions of five functions of the ingredients (continuous variables): high polarity emollient, medium polarity emollient, low polarity emollient, UVA filter, UVB filter.
- UV filter ratios (continuous variables): UVB filter vs UVA filter (+broad spectrum filter), UV filter in the oil phase vs UV filter in the water phase and absorbing type UV filter vs scattering/reflecting type UV filter.
- A description of the UV protection of the final product (continuous variables): SPF *in vivo* and UVA-PF *in vitro*.

A summary of the dataset can be found in Table 4.2.

Table 4.2 Summary of all types sunscreen product dataset.

	Outputs		Inputs								
	SPF in_vivo	UVA-PF *in vitro*	Visocosity (mPa*s)	Emollient high	Emollient low	Emollient medium	Scattering /Absorber in UV filters	UV filter in water	UVA/ UVB	UVB filter	UVA filter
count	261	261	261	261	261	261	261	261	261	261	261
mean	**40.80**	16.08	19081.53	16.79	10.40	2.79	0.22	0.28	0.62	6.99	4.96
std	**16.96**	6.67	19118.35	63.13	64.65	5.56	0.28	0.26	0.58	5.31	5.05
min	**4.10**	3.00	50.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	**29.00**	10.60	4900.00	5.00	2.00	0.00	0.00	0.00	0.19	2.40	1.00
50%	**35.00**	16.00	15000.00	10.00	4.00	0.00	0.10	0.26	0.49	6.50	5.00
75%	**55.00**	20.70	27000.00	19.00	8.00	4.00	0.41	0.47	0.91	11.50	7.00
max	**90.00**	38.60	105000.00	1010.00	1042.00	49.50	1.00	0.84	3.49	22.50	64.30

4.4 Results and Discussion

4.4.1 O/W type sun cream product

Feature importance

By applying the feature engineering methods introduced in Section 4.2.1 to the O/W dataset described in Section 4.3, the direct outcome would be the ranking of features importances.

The features importances ranking under different feature engineering methods are shown in the figures below. In Fig. 4.2, the features are ranked according to Pearson's r values, which correspond with the linear correlations between features and outputs. It can be seen that the UVA filter and UVB filter concentration have a high positive linear correlation with the SPF and UVA values. This matches the expert's domain knowledge. It should also be notice that all the features have a more then ±0.1 linear correlation to the sun protection value, which means they all have a contribution to the final results. This further supports the argument that sun cream protection is jointly, to some extent evenly, influenced by various features; which proofs the complexity of design of sun cream product. By applying random forest, as shown in Fig. 4.3, viscosity was identified as one of the most important features, which is also consistent with expert's observation. The identified can be selected and then used to build machine learning models, which can reduce the model dimensionality and therefore improve the prediction accuracy.

Feature Space Visualization

To better illustrate how feature engineering help ML modeling, some feature space visualization effort is shown in this section. Although it is quite difficult to visualize the feature distribution in high dimensional space, there are still several useful techniques that could give

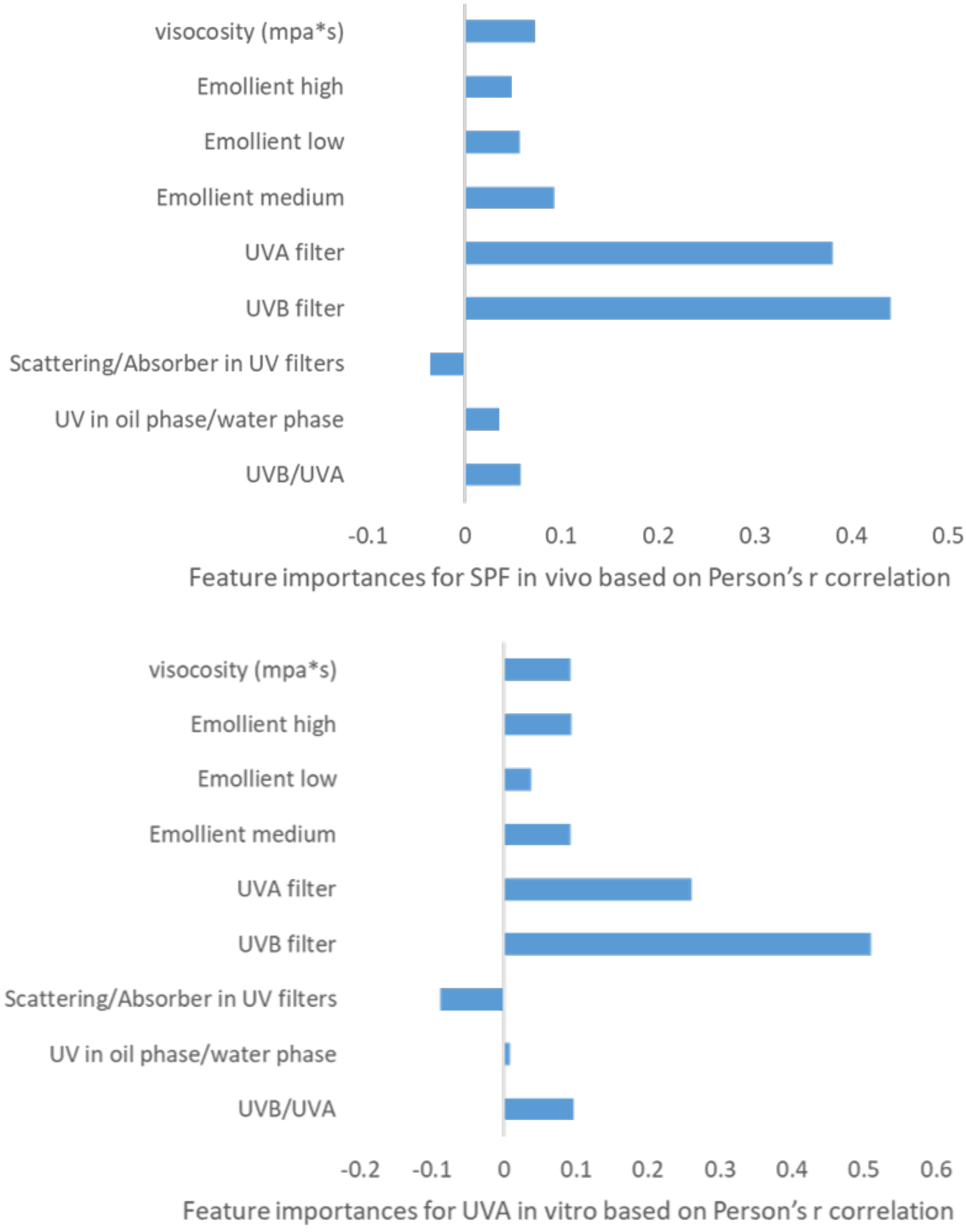
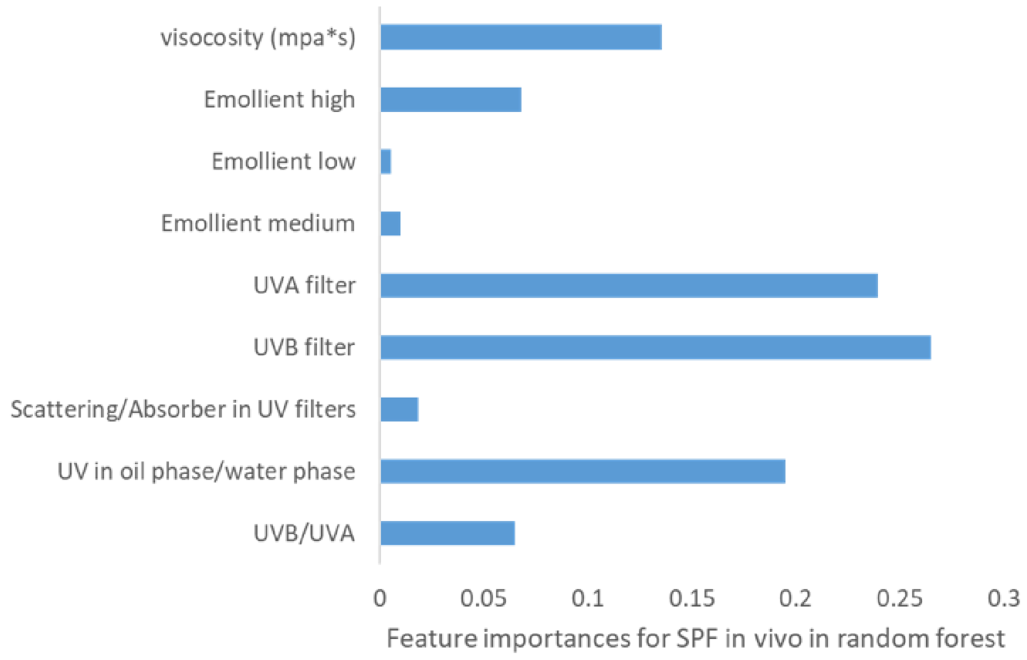


Fig. 4.2 Feature importances for SPF and UVA based on Person's r correlation.



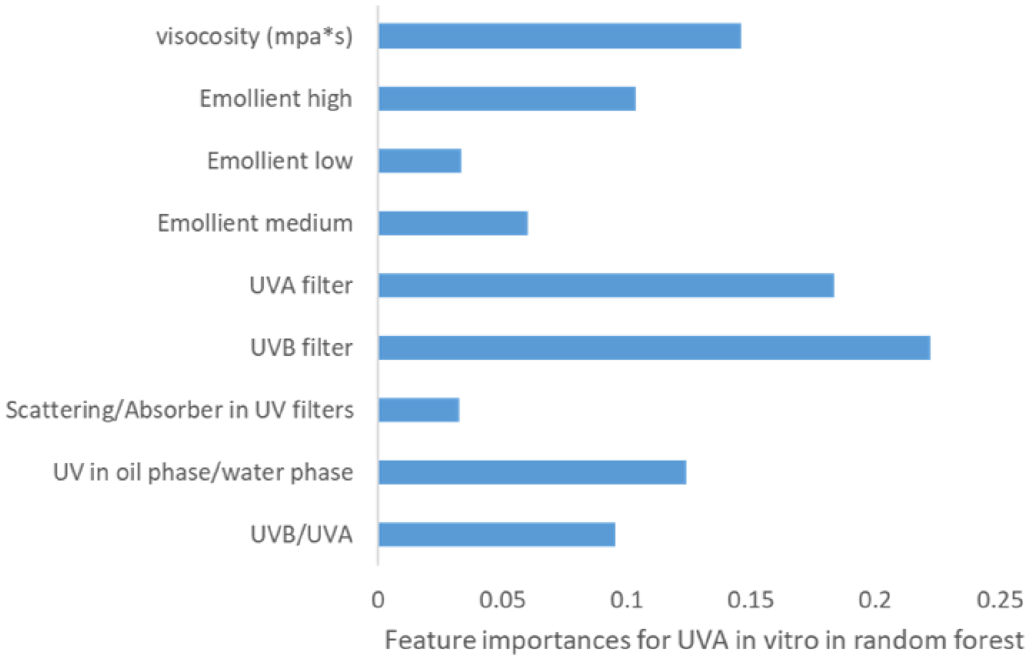


Fig. 4.3 Feature importances for SPF and UVA based on Random forest method.

us some fundamental insights. Pairwise scatter plot is one of them. In a pairwise scatter plot, the features are plotted against output such that the one-dimensional distribution of output in the feature space can be obtained. In Fig. 4.5, pairwise scatter plot in the original feature space is shown; only the five most important features in the original feature space are shown. In Fig. 4.6, pairwise scatter plot in the transformed feature space through PCA is shown, again only the five most important features are explored in the plot. Principal Components in Fig. 4.4 are non- dimensional variables in the new feature space without any real physical meaning. Although there is still no obvious pattern in this figure, compared to the original feature space, the distribution has been flattened, which would make it easier to develop more efficient predictors.

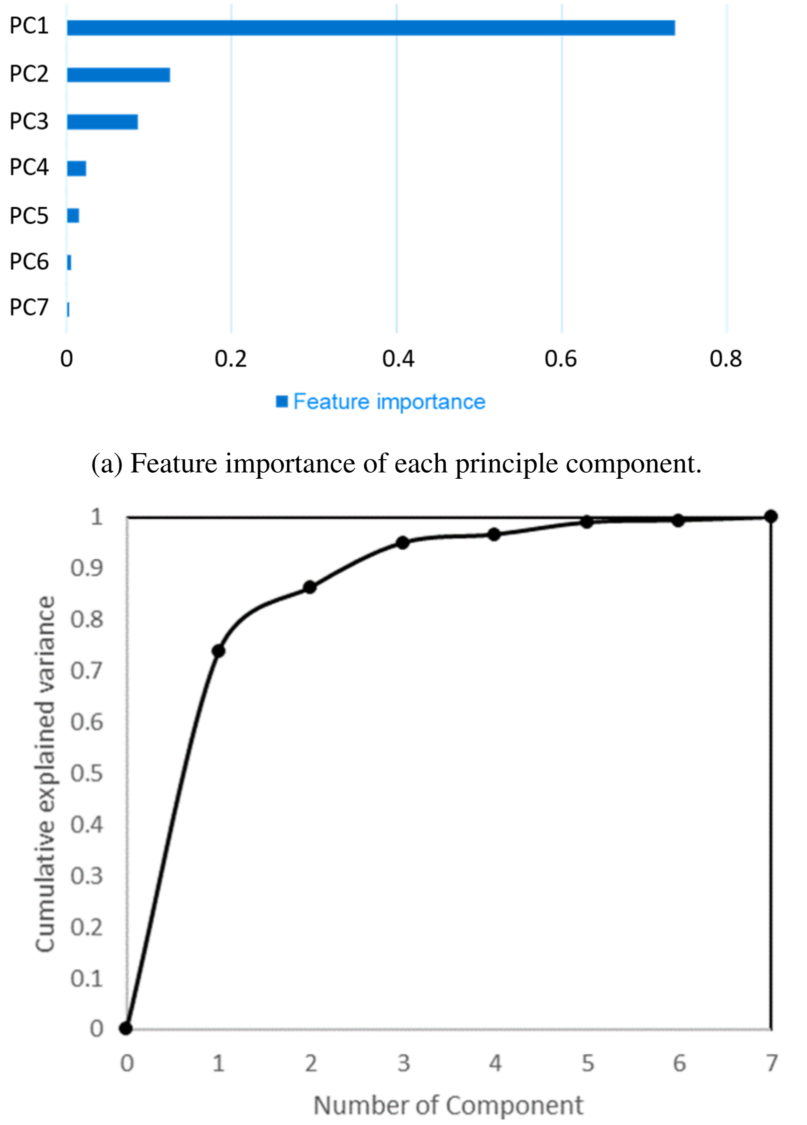
Model Fitting

The mean performance of each type of the model was reported in Table 4.3. It is defined as mean square error between the model prediction and the experimental data. It can be seen that Bayesian regression has the best performance among the six different models. Moreover, as it is shown in Table 4.4, by applying feature engineering method, the prediction accuracy was improved.

4.4.2 Influence of Different Types of Sunscreen Products

Encoding Discrete Variables

Categorical data is common in experimental datasets, however, it is hard to be used in machine learning algorithms, due to the fact that many machine learning models are algebraic, thus their input must be numerical. Therefore, to use these models, categorical features must be transformed into continues numbers. Analysis and pre-processing of mixed datasets including a combination of continuous can categorical variables has an important research interest



(a) Feature importance of each principle component.

(b) Cumulative explained variance by different number of principle components.

Fig. 4.4 Feature importance with principle component analysis.

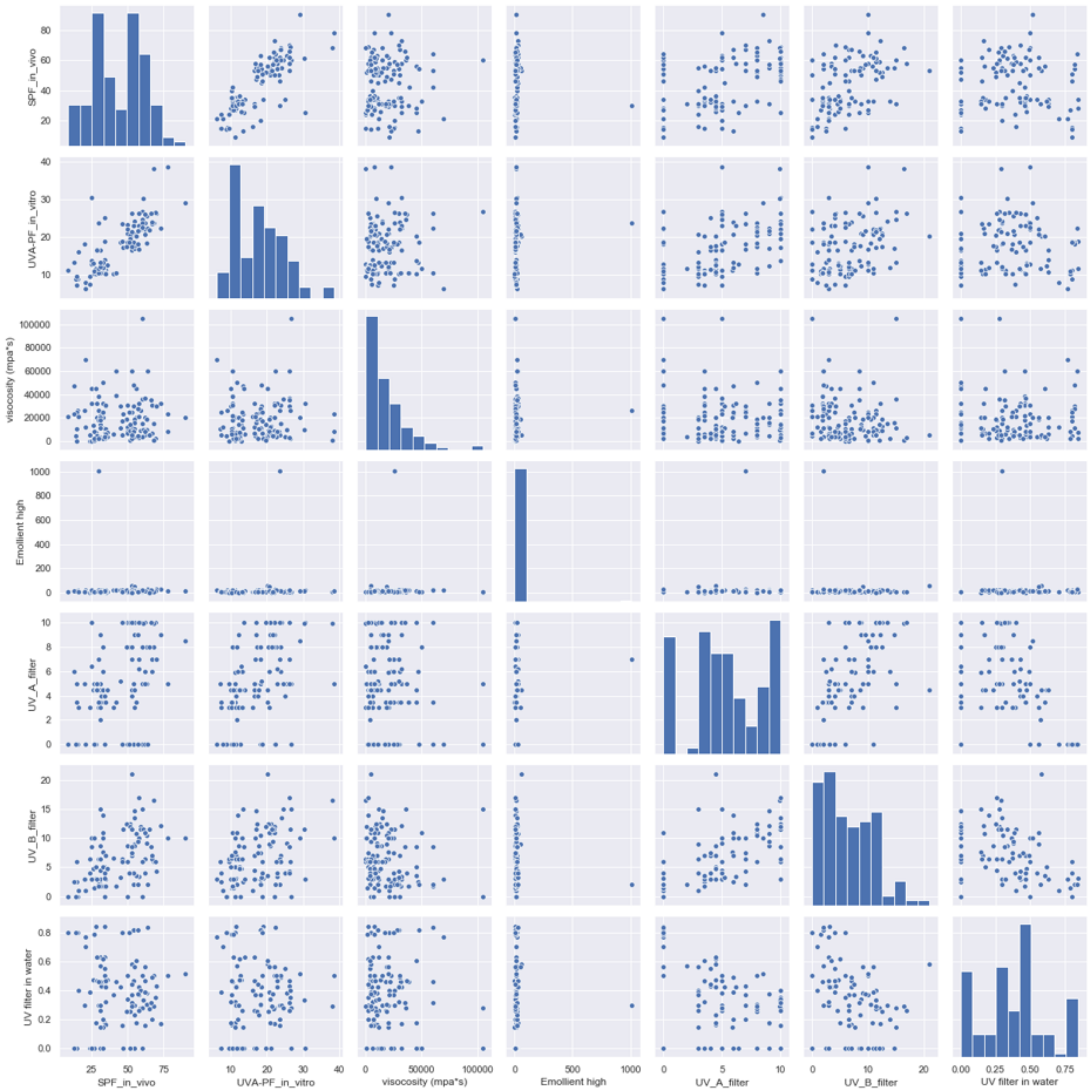


Fig. 4.5 Scatter plot between SPF/UVA and the five most important features in original feature space.

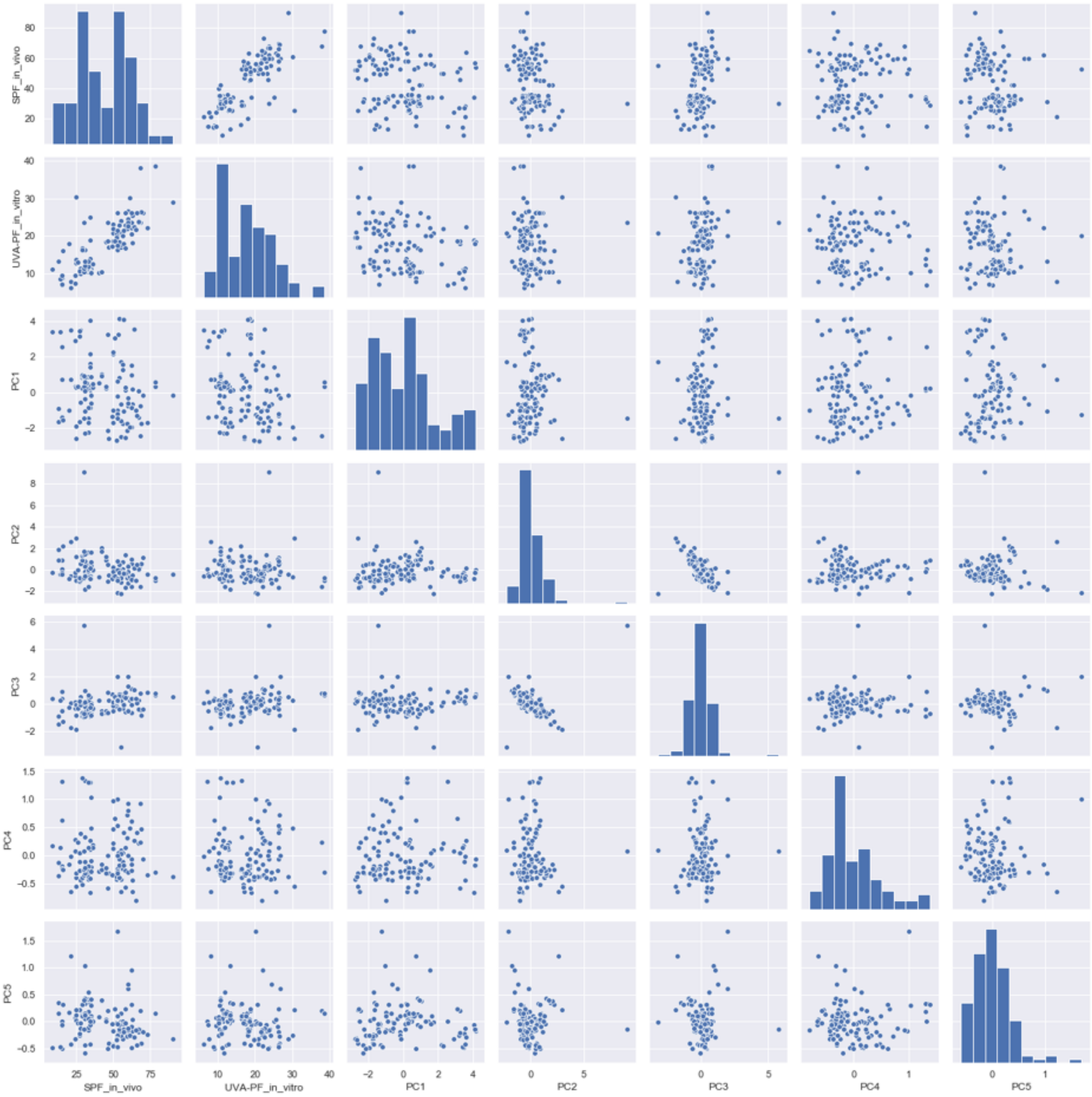


Fig. 4.6 Scatter plot between SPF/UVA and the five most important features in PCA-transformed space.

Table 4.3 Mean performance of six different types of models in SPF and UVA prediction.

SPF *in vivo* prediction					
Ridge Regression	Bayesian Regression	SVM	k-NN	Decision Tree	GP
36.28	18.10	20.89	26.32	28.83	45.99
UVA-PF *in vitro* prediction					
Ridge Regression	Bayesian Regression	SVM	k-NN	Decision Tree	GP
26.85	15.10	18.29	19.51	19.77	28.97

Table 4.4 Mean performance of 6 different types of models in SPF and UVA prediction.

SPF *in vivo* prediction		
Entire feature space	5 most important features	5 most important PCs
18.10	15.18	14.97
UVA-PF *in vitro* prediction		
Entire feature space	5 most important features	5 most important PCs
15.10	10.72	9.25

in the past decade [207, 208]. There are multiple methods which can convert categorical features into continuous variables, such as encoding to ordinal variables, feature hashing, Cat2vec and others. Due to the fact a small dataset was provided, I only applied the methods which usually works well within small dataset, namely encoding to ordinal variables, one hot encoding and feature hashing [209]. Performance was evaluated through the prediction accuracy based on Bayesian regression model. As shown in Table 4.5, the method which encodes the sun cream product type to ordinal variables has the lowest prediction error; therefore the encoding method works the best compare to the rest of two.

Table 4.5 SPF and UVA prediction model performance using different category encoding methods.

SPF *in vivo* prediction		
Encoding to ordinal variables	One hot encoding	Feature hashing
28.87	39.40	37.54
UVA-PF *in vitro* prediction		
Encoding to ordinal variables	One hot encoding	Feature hashing
20.26	35.64	28.30

Feature Importance Analysis

Based on the encoding to ordinal variables method, feature importances were analysed to understand which features have larger impact in the entire sun cream product database. It can be noticed from Fig. 4.7, product type is the most important feature. Also, compared to o/w and gel type product, similar knowledge can be obtained when we use the whole product database. Viscosity and UVB filter concentration is within the top three important features.

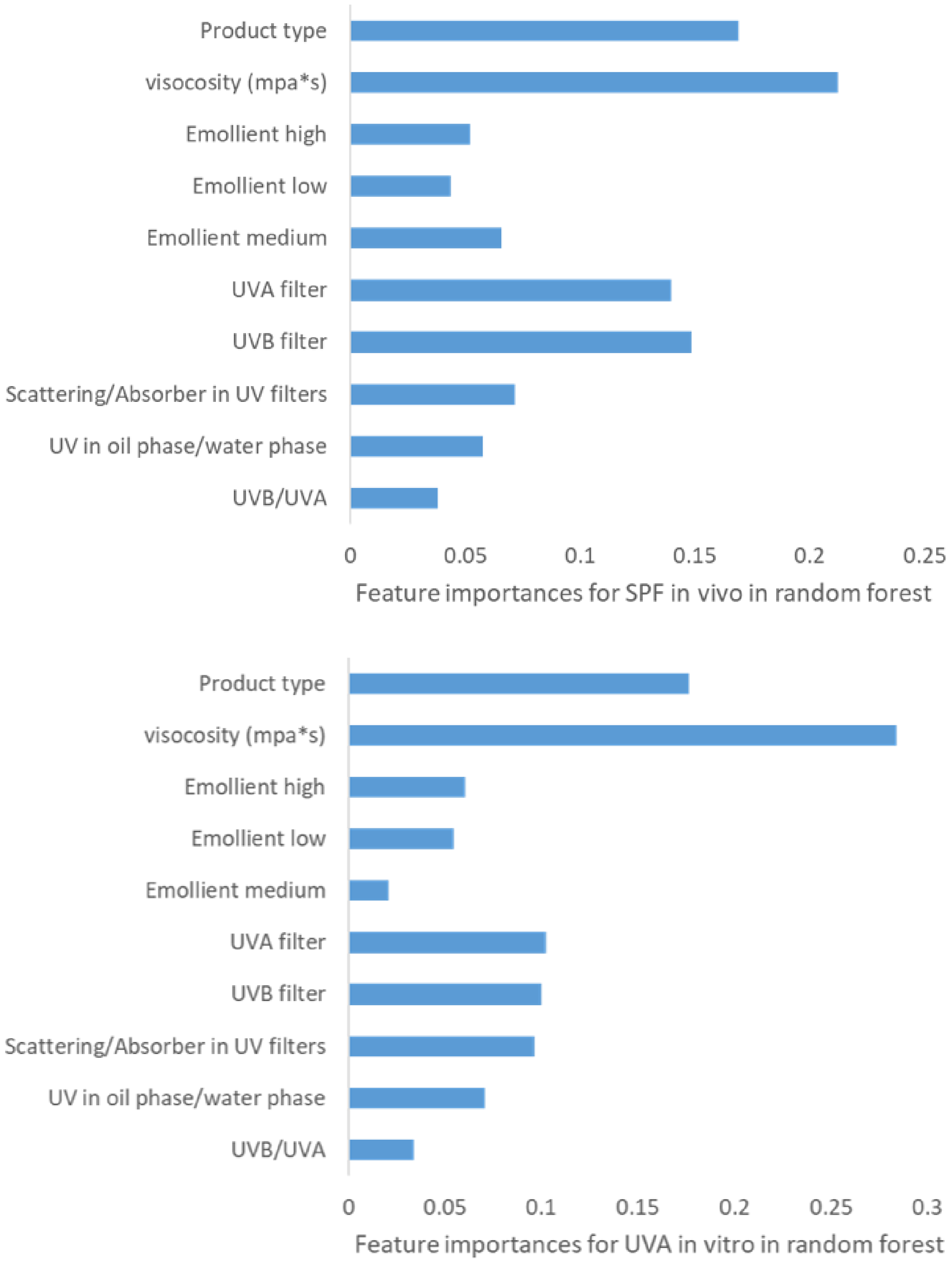


Fig. 4.7 Feature importance analysis using random forest for the full dataset.: (a) Feature importance for SPF *in vivo* in random forest (b) Feature importance for UVA-PF *in vitro* in random forest.

4.5 Conclusions

In this work, feature engineering in sun cream protection prediction is discussed. Three different types of feature engineering methods, namely feature visualization, feature selection, and feature extraction are discussed in the chapter. Exploratory Data Analysis (EDA), random forest (RF), and Principal Component Analysis (PCA) are used to implement feature visualization, feature selection, and feature extraction respectively. By applying such methods to the o/w and gel type database, some insights about the feature space can be obtained. Concentrations of UVA and UVB filters are identified as the key features, as well as the product viscosity. This matches with the domain knowledge experts have in sun cream product design, although the fundatmental reason for the effect of viscosity is unknown. Feature extraction methods, such as PCA can transform the original feature space into a new principal component space within which the machine learning models may be easier to develop. Six types of different machine learning models were applied to the dataset. Bayesian regression model achieved the least prediction error and is the most promising for further development. By applying feature engineering method before fitting in the ML model, it gives higher prediction accuracy, compared to the previous attempt of using ML method for SPF and UV prediction [193]. Moreover, the ML developed within this can give comparable prediction accuracy to the one based on physical models [191], but with wider application scenario over different UV protectors and emollients in our case.

The entire database of the sun cream products was also analysed. To convert the categorical product type into numerical features, three methods were applied. The one which encode categorical into ordinary variables has the least prediction error in Bayesian regression model, which suggests this is the best encoding method so far for the dataset. Based on this method, feature importances were analysed as well. Similarly, it can be seen that product viscosity and UVB filter concentration are crucial to SPF *in vivo* and UVA-PF *in vitro* values. Moreover, product type is also vital to for sun cream products, which matches to experts' knowledge.

Chapter 5

Identifying physico-chemical laws from experimental data using symbolic regression

5.1 Introduction

In the previous chapters, we have shown that by combining machine learning and robotic systems, digital molecular technology promises to significantly expand the accessible chemical space, and to reduce the price of access to new functional molecules and materials. The key component of the new Digital Molecular Technology (DMT) methods is the increased volume and quality of chemical data obtained both through data mining, computational chemistry tools and robotic experiments, as lack of data renders ML and AI methods inaccurate and not very useful [210]. Here we ask a question, *is it possible to make use of the increased availability of experimental data to enhance our capability in inferring physical knowledge from data by means of algorithmic research?* This is driven by the desire to develop predictive models of complex chemical processes, which could later be used in optimal control. The approach that we seek to develop should, ideally, not be based on selecting functional forms from a pre-defined set. This has already been demonstrated within DMT, for example, in selecting suitable kinetic expressions within an automated self-optimization system [211]. Our own interest is in the methods that are inherently not restricted to only the known functional forms and are, therefore, potentially capable of discovering new physical models.

The data-driven development of interpretable physical models resisted automation for a long time [164]. The possibility of building data-driven interpretable and generalizable models for complex and poorly understood physical systems is important as these models share a similar structure to those, based on first principles, and can be transferred to analogous systems, whereas surrogate models cannot be easily generalized [212]. This, in the longer term, can improve the time and resource efficiency for the product discovery and process optimization, especially for the manufacturability and the scale-up, for which a mechanistic understanding is often crucial [48, 211, 213].

The field of algorithmic search for physical models is relatively new, but has seen a number of important advances. There are two main types of methods in this field: parametric and non-parametric regression. In parametric regression, the potentially nonlinear functional form is known *a priori* or approached by a weighting of multiple known basis functions. As a parametric approach, ALAMO (automated learning of algebraic models for optimization) platform can identify surrogate models from small datasets that are as accurate and as simple as possible [214]. In [214], ALAMO was extended to incorporate a priori physical knowledge by enforcing physical constraints on the model resulting from parametric regression; this was further illustrated through application of ALAMO to learning problems in kinetics [143]. Similar approaches that linearly combine the candidate functions from a pre-defined library are numerous in the literature domain of sparse regression. The identification of a data-driven physical model (DDPM), specifically of sparse and interpretable (partial) differential equations of nonlinear dynamical systems, has been successful with parametric regression techniques [215–219]. For instance, the technique was successfully applied to the data-driven discovery of Navier-Stokes equations [218]. Furthermore, sparse regression was extended to include selection of candidate basis functions by dimensional analysis, and by determining the parameters including error bars by a hierarchical Bayesian framework [220].

In non-parametric regression, the *a priori* selection of basis functions is not needed. Therefore, the non-parametric methods allow us to find free form equations. Recently, Brunton et al. introduced a sparse regression approach to discover equations governing the physics in a chaotic Lorenz system, and in a fluid vortex dynamic system [221]. However, this technique is restricted to a pre-defined algebraic model structure, as selected basis functions are linearly combined. Allowing free-form analytical equations, Bongard and Lipson developed a criterion to find meaningful and complex mathematically invariant models by means of the ML method of symbolic regression (SR) [222].

Recent applications of SR for physical models can be found in civil engineering [223] and material science [224]. Although successfully proven, the earlier proposed SR was based on a heuristic search that could terminate the optimization in local minima solutions, potentially producing less suitable models than possible. Additionally, as the structure of a model reflects the actual mechanistic interactions within the system studied, these approximate solutions cannot be used reliably to infer any mechanistic information about the system, i.e. to use it to identify chemical reaction mechanisms with certainty. Acknowledging this disadvantage, SR is formulated as a mixed-integer nonlinear programming (MINLP) in Refs [225, 226] and solved to global optimality. However, until now the method remains in the mathematical domain and is yet to be applied to physical models and noisy experimental data.

Here we aim to advance the method of globally optimal symbolic regression towards automated, data-driven identification of physical models, and its applications to chemical engineering case studies. Compared to additive models in conventional regression and heuristic searches in SR, the globally optimal data-driven modelling technique, without any previously imposed model structure, is expected to discover true underlying relationships more reliably. To accomplish this, in this chapter a modified optimization formulation of SR is developed and implemented in combination with a framework for physical model selection. As proof of concept, several case studies were investigated in the areas of rheology and reaction engineering. The purpose of this work is to illustrate an automated research pipeline deriving interpretable and generalizable models, and thereby providing access to physical knowledge from data. Within this big picture, closing the loop of utilizing the obtained physical models in further experimentation and generation of physical knowledge by (automated) interpretation, see Fig. 5.1, remains the target for future research. The caveat to this is that we cannot expect such a methodology to be able to discover new phenomena for which no physical response is measured. The assumptions for the model are that all predictor variables are included in measurements, all possible operators are included and the tree structure is large-enough to be able to discover the physical phenomena of a sufficiently complex system.

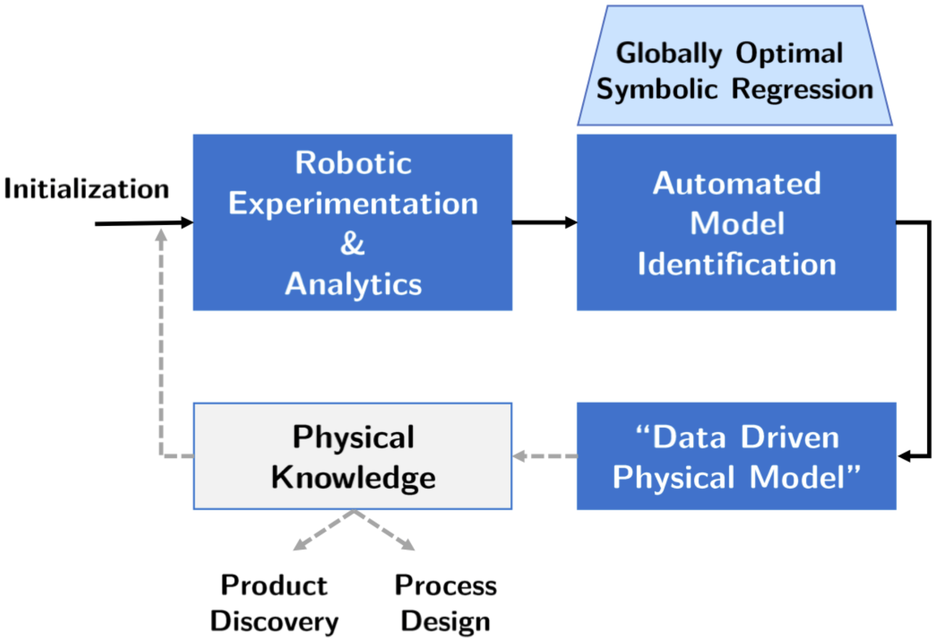

Globally Optimal Symbolic Regression
Initialization
Robotic Experimentation & Analytics
Automated Model Identification
"Data Driven Physical Model"
Physical Knowledge
Product Discovery
Process Design

Fig. 5.1 Schematic diagram of a concept of a closed loop automated physical model identification.

5.2 Materials and Methods

5.2.1 MINLP formulation

The MINLP formulation is based on a balanced binary tree superstructure for representation of the equations describing a physical model. The overall goal is to enable the assembly of free-form algebraic functions by connecting predictor variables and operators, such that the resulting function predicts the dependent variable values accurately. As an example, the structure with nodes in a four-layer balanced binary tree is shown in Fig. 5.2.

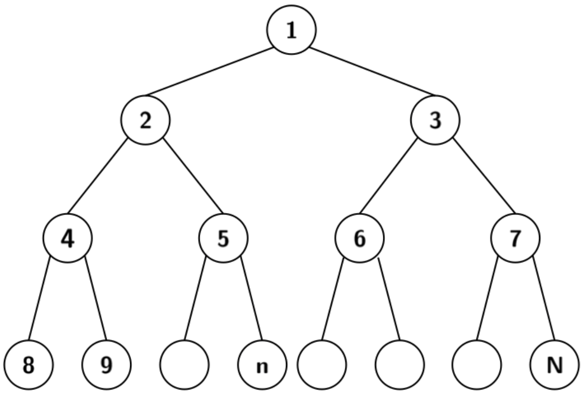


Fig. 5.2 An example of a four-layer balanced binary tree.

The formulation presented here is based on Ref. [226], but follows a different concept in the set-up of the binary tree, which allows us to reduce the number of binary variables in the global optimization. These modifications are addressed in Section 5.2.3.

An expression tree consists of $N = 2^{NL} - 1$ nodes, where NL defines the number of layers. All nodes that have a connection with nodes on a lower level, their child nodes, are called branch-nodes ($\mathcal{N}_b$) or non-leaf nodes, and house a mathematical operator. The nodes on the lowest layer in the tree, referred to as leaf-nodes ($\mathcal{N}_l$), are assigned to a predictor variable ($x_{i,j}$) or a constant (ε_n). In the following sections we will refer to the total number of activated nodes in the tree as "complexity of the model" (except the ones with an identity operator). Each given data point deployed in the SR is described by two parameters: the value of the predictor variable ($x_{i,j}$) and the dependent variable value (y_j), which is predicted by the model for each training data point (j). As shown in Fig. 5.3, the input variables are assigned for selection at the leaf-nodes only, while the dependent variable values are used at the root node ($n = 1$) for comparison with the model prediction.

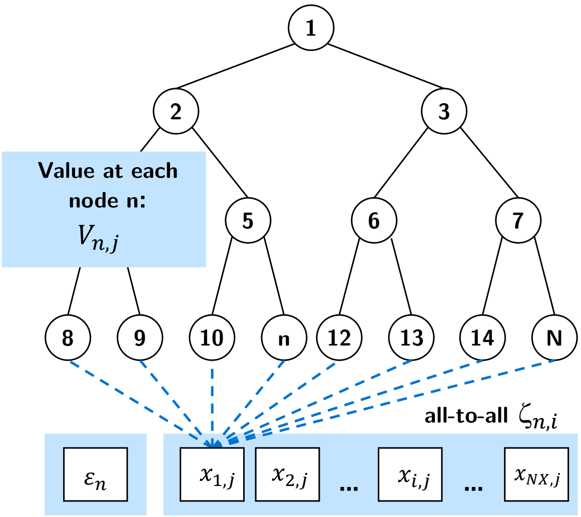


Fig. 5.3 The MINLP set-up in connection with the expression tree.

Each node has a value for each data point ($V_{n,j}$), which is computed to be used as operator arguments on the layer above. The nodal values at the bottom of the tree are determined by the selection of an input variable or a constant. On branch node layers, nodal values are specified by the selected operator in combination with the node values of their children. The allocation of the input predictor variables ($x_{i,j}$) is implemented by the binary variables ($\zeta_{n,i}$). Continuous decision variables (ε_n) with bounds ($\varepsilon_n^l o$) and ($\varepsilon_n^u p$) are designated for constants at every even leaf-node. To decide between a variable input and the selection of a constant at the even leaf-nodes, further binary variables (κ_n) are assigned. Both (κ_n) and (ε_n) are only assigned to the even leaf-nodes, as the left leaf-nodes in each last bifurcation only contain the constant values. With regard to the branch-nodes, there are binary variables ($\delta_{n,k}$) assigned for operator selection, where an operator is active at node n if $\delta_{n,k} = 1$ and inactive if $\delta_{n,k} = 0$. If active, each binary operator is applied using both child nodes while a unary operator uses only the value of the left node ($V_{2n,j}$). Five binary operators (addition, subtraction, multiplication, division and power law) and three unary operators (identity, exponential and square root) were implemented. It should be noted that the list of operators can be extended further, such as cubic, square or logarithm operations as proven in Ref. [226].

Consequently, by using the tree structure and the index assignment (Table 5.1 to Table 5.3), the optimization problem was formulated with the objective to minimize the sum of squared errors (SSE) between the values computed by the model and the experimental data (Eq. 5.1), according to Ref. [226].

Eqs. (5.2) to (5.4) enable the operator selection at branch-nodes with a big-M approach. With the aim of connecting the nodal values in the hierarchy of the tree, the functions (f_k) are introduced for each operator, as shown in Table 5.4. If additional requirements apply for the selection of an operator, such as a non-zero denominator for division or a positive base for a power law, these are provided in the functions (g_k) enforcing a minimum distance to undefined regions. A detailed explanation of the implementation of big-M approach can be found in Refs.[226] and [227]. The idea of the big-M facilitates the transformation of the disjunctive choice between the operators into linear constraints [227]. If $\delta_{n,k} = 1$, the two inequality constraints reduce to $f_k = 0$, which enforces the respective mathematical operation. In contrast, if $\delta_{n,k} = 0$, large M-values, as lower and upper bounds, ensure the free selection of nodal values within their specific bounds. The coefficients ($M^{up}_{n,j,k}$), ($M^{lo}_{n,j,k}$) and ($G^{up}_{n,j,k}$) were introduced to preserve constraint linearity in the binary decision variables, which is important to reduce non-convexities and for solving the MINLP efficiently.

$$min \sum_{j=1}^{NE} (y_j - V_{1,j})^2 \tag{5.1}$$

$$f_k(V_{n,j}, V_{2n,j}, V_{2n+1,j}) \leq M^{up}_{n,j,k}(1 - \delta_{n,k}), \quad n \in \mathscr{N}_b, k \in \mathscr{F}, j \in \mathscr{J} \tag{5.2}$$

$$f_k(V_{n,j}, V_{2n,j}, V_{2n+1,j}) \geq M^{lo}_{n,j,k}(1 - \delta_{n,k}), \quad n \in \mathscr{N}_b, k \in \mathscr{F}, j \in \mathscr{J} \tag{5.3}$$

$$g_k(V_{n,j}, V_{2n,j}, V_{2n+1,j}) \leq G^{up}_{n,j,k}(1 - \delta_{n,k}), \quad n \in \mathscr{N}_b, k \in \mathscr{F}, j \in \mathscr{J} \tag{5.4}$$

$$V_{n,j} \leq V^{up}_{n,j} \sum_{k \in \mathscr{F}} \delta_{n,k}, \quad n \in \mathscr{N}_b, j \in \mathscr{J} \tag{5.5}$$

$$V_{n,j} \leq V^{lo}_{n,j} \sum_{k \in \mathscr{F}} \delta_{n,k}, \quad n \in \mathscr{N}_b, j \in \mathscr{J} \tag{5.6}$$

$$\sum_{k \in \mathscr{F}} \delta n, k \leq 1, \quad n \in \mathscr{N}_b \tag{5.7}$$

$$V_{n,j} = \sum_{i \in \mathscr{X}} \zeta_{n,i} x_{i,j} + \kappa_n \varepsilon_n, \quad n \in \mathscr{N}_l{}^*, j \in \mathscr{J} \tag{5.8}$$

$$V_{n,j} = \sum_{i \in \mathscr{X}} \zeta_{n,i} x_{i,j}, \quad n \in \mathscr{N}_l \setminus \mathscr{N}_l{}^*, j \in \mathscr{J} \tag{5.9}$$

$$\sum_{i \in \mathscr{X}} \zeta_{n,i} + \kappa_n \leq 1, \quad n \in \mathscr{N}_l^* \tag{5.10}$$

$$\sum_{i \in \mathscr{X}} \zeta_{n,i} \leq 1, \quad n \in \mathscr{N}_l \setminus \mathscr{N}_l^* \tag{5.11}$$

$$\sum_{n \in \mathscr{N}_l} \sum_{i \in \mathscr{X}} \zeta_{n,i} \geq 1 \tag{5.12}$$

$$\delta_{n,k} \in \{0, 1\}, \quad n \in \mathscr{N}_b, k \in \mathscr{F} \tag{5.13}$$

$$\zeta_{n,i} \in \{0, 1\}, \quad n \in \mathscr{N}_l, i \in \mathscr{X} \tag{5.14}$$

$$\kappa_n \in \{0, 1\}, \quad n \in \mathscr{N}_l^* \tag{5.15}$$

$$V_{n,j} \in [V^{lo}_{n,j}, V^{up}_{n,j}], \quad n \in \mathscr{N} \tag{5.16}$$

$$\varepsilon_{n,j} \in [\varepsilon^{lo}_n, \varepsilon^{up}_n], \quad m \in \mathscr{N}_l^* \tag{5.17}$$

If no operator is selected, its nodal value is set to zero by constraints (Eqs. (5.5) and (5.6)).

Table 5.1 MINLP Notation: Set.

Description	Index	Set	Value
Nodes	n	$\mathscr{N}$	$\{1, \cdots, N\}$
Branch-nodes		$\mathscr{N}_b$	$\{1, \cdots, 2^{NL-1}-1\}$
Leaf-nodes		$\mathscr{N}_l$	$\{2^{NL-1}, \cdots, N\}$
Even leaf-nodes		$\mathscr{N}_l^*$	$\{2^{NL-1}, 2^{NL-1}+2, \cdots, N\}$
Algebraic Operators	n	$\mathscr{F}$	$\{+, -, \cdots\}$
Predictor Variables	i	$\mathscr{X}$	$\{1, \cdots, NX\}$
Input Data Points	j	$\mathscr{J}$	$\{1, \cdots, NE\}$

Table 5.2 MINLP Notation: Parameter.

Description	Parameter	
Input variable values	$x_{i,j}$	$i \in \mathscr{X}, j \in \mathscr{J}$
Dependent variable values	y_j	$j \in \mathscr{J}$

Table 5.3 MINLP Notation: Decision variables.

Applicability	Description	Variable		Bounds
General	Nodal values	$V_{n,j}$	$n \in \mathscr{N}_b, j \in \mathscr{J}$	$[V^{lo}_{n,j}, V^{up}_{n,j}] \in \mathbb{R}$
Leaf-nodes	Variable selection	$\zeta_{n,i}$	$n \in \mathscr{N}_l, i \in \mathscr{X}$	{0,1}
	Constant selection	κ_n	$n \in \mathscr{N}_l^*$	$\{0,1\}$
	Value of constants	ε_n	$n \in \mathscr{N}_l^*$	$[\varepsilon^{lo}_{n,j}, \varepsilon^{up}_{n,j}]$
	Operator selection	$\delta_{n,j}$	$n \in \mathscr{N}_b, k \in \mathscr{F}$	$\{0,1\}$

Table 5.4 MINLP Notation: Operator Implementation.

	Description	Index	f_k	g_k
Binary Operators	Addition	+	$V_{2n,j} + V_{2n+1,j} - V_{n,j}$	
	Subtraction	-	$V_{2n,j} - V_{2n+1,j} - V_{n,j}$	
	Multiplication	$\cdot$	$V_{2n,j} \cdot V_{2n+1,j} - V_{n,j}$	
	Division	/	$V_{2n,j} - V_{2n+1,j} \cdot V_{n,j}$	$\varepsilon - V_{2n+1,j}^2$
	Power Law	$\wedge$	$e^{ln(V_{2n+1,j})V_{2n,j}} - V_{n,j}$	$\varepsilon - V_{2n_1,j}$
Unary Operator	Identity	*id*	$V_{2n,j} - V_{n,j}$	
	Exponential	exp	$e^{V_{2n,j}} - V_{n,j}$	
	Square Root	$\sqrt{\quad}$	$V_{2n.j} - V_{n,j}^2$	$\varepsilon - V_{2n,j}$

Additionally, either none or one operator can be selected at the branch-nodes. Hence, the sum of operator binaries must be less or equal to one, which is constrained by Eq. (5.7).

In contrast to the branch-node values, the values at the leaf-nodes are determined by equality constraints including the binary selection of predictor variables or constants, Eqs. (5.8) and (5.9). Also, Eqs. (5.10) and (5.11) make sure that either no operand, one variable or one constant can be assigned. Overall, the model should include at least one predictor variable, which is ensured by Eq. (5.12). For the purpose of completeness, Eqs. (5.13) to (5.17) depict the bounds on decision variables of the MINLP [226].

$$\sum_{k\in\mathscr{F}_{unary}} \delta_{n,k} \leq 1 - \sum_{k\in\mathscr{F}} \delta_{2n+1,k}, \quad n \in \{1,\cdots,2^{NL-2}-1\} \tag{5.18}$$

$$\sum_{k\in\mathscr{F}_{unary}} \delta_{n,k} \leq 1 - \sum_{i\in\mathscr{X}} \zeta_{2n+1,i}, \quad n \in \{2^{NL-2},\cdots,2^{NL-1}-1\} \tag{5.19}$$

$$\sum_{k\in\mathscr{F}} \delta_{n,k} \geq \sum_{k\in\mathscr{F}} \delta_{2n,k}, \quad n \in \{1,\cdots,2^{NL-2}-1\} \tag{5.20}$$

$$\sum_{k\in\mathscr{F}} \delta_{n,k} \geq \sum_{k\in\mathscr{F}} \zeta_{2n+1,k}, \quad n \in \{1,\cdots,2^{NL-1}-1\} \tag{5.21}$$

$$\sum_{k\in\mathscr{F}} \delta_{n,k} \geq \sum_{i\in\mathscr{X}} \zeta_{2n,i} + \kappa_n, \quad n \in \{2^{NL-2},\cdots,2^{NL-1}-1\} \tag{5.22}$$

$$\sum_{k\in\mathscr{F}} \delta_{n,k} \geq \sum_{i\in\mathscr{X}} \zeta_{2n+1,i}, \quad n \in \{2^{NL-2},\cdots,2^{NL-1}-1\} \tag{5.23}$$

$$V_{2n,j'} - V_{2n+1,j'} \geq M^{SC}_{n,j'}(1 - \sum_{k=\{+,*\}} \delta_{n,k}), \quad n \in \mathscr{N}_b, \exists j' \in \mathscr{J} \tag{5.24}$$

Due to the binary architecture and the commutative nature of addition and multiplication, the expression tree contains many mathematically invariant models (symmetries). The design of the formulation should, therefore, impede redundancies. Eqs. (5.18) and (5.19) resemble cuts in the tree such that, if a unary operator is selected, the unused part towards the right child node is set to zero [226]. Eqs. (5.20) to (5.23) assure that if an operator is selected on a lower layer of the expression tree, there is an operator attached to the parental node [226]. Likewise, it ensures that the children of a node with value zero also have no operator or variables attached.

Additionally, symmetry breaking cuts (SC) to remove redundant solutions, which are caused by the commutative nature of addition and multiplication, were implemented. Eq. (5.24) is sufficient for one data point $j = j'$ to impose an order on the values of the child nodes [226]. The symmetry breaking cuts also pose as big-M constraints where M_n^* is set using interval arithmetic on the bounds of the two child node values [227].

Without a doubt and from experience with big-M formulations in the Mathematical Programming community, this will lead to rather loose lower bounding in the associated Branch-and-Bound (B&B) traditionally used to solve Mixed-Integer Linear (MILP) and Mixed-Integer Nonlinear Programming (MINLP) problems. Indeed, such were the observations reported later in the computational results of this work, and hence the serious limitations that is a challenge for future development of this rigorous methodology.

5.2.2 Determination of Big-M Coefficients

For the optimization, the MINLP variable bounds as well as appropriate big-M values must be provided. Keeping them as small as possible is expected to reduce convergence times in solving the MINLP. In the following, an automatable approach for any supplied dataset is proposed based on bottom-up interval arithmetics on the training data itself. Initiated at the leaf-nodes, the node value bounds $V_{n,j}^{lo/up}$can be determined based on the maximum and minimum values within the dataset plus-minus a safety margin ρ. At even leaf-nodes the bounds of constants are considered (Eqs. (5.25) and (5.26)). The bounds on the constants $\varepsilon_n^{lo/up}$ are defined *a-priori*.

$$V_{n,j}^{lo} = \min[\varepsilon_n^{lo}, x_{i,j} \forall i] - \rho, \quad j \in \mathscr{J}, n \in \mathscr{N}_l^* \tag{5.25}$$

$$V_{n,j}^{up} = \max[\varepsilon_n^{up}, x_{i,j} \forall i] - \rho, \quad j \in \mathscr{J}, n \in \mathscr{N}_l^* \tag{5.26}$$

For the next layer above and subsequently for all other branch-nodes, the overall nodal value bounds $V_{n,j}^{lo/up}$are determined as maximum or minimum value of interval arithmetics based nodal bounds $V_{n,j,k}^{lo/up}$ calculated for each operator and training data point (Eqs. (5.27) and (5.28)). The calculation of operator specific $V_{n,j,k}^{lo/up}$ is defined in Table 5.5. For improved readability in the table, the j-indices are omitted but must be considered in the calculation. As the intervals grow quickly with the number of layers, a predefined nodal value limit V_{limit}

can be set. Moreover, the root node bounds are determined top-down from the values of the independent variable to be predicted. A model that remains within two standard deviations 2σ of the expected value is proposed in Eq. (5.29).

$$V^{lo}_{n,j} = \min[-V_{limit}, V^{lo}_{n,j,k}\ k \in \mathscr{F}], \quad j \in \mathscr{J}, n \in \mathscr{N}_b \tag{5.27}$$

$$V^{up}_{n,j} = \min[-V_{limit}, V^{up}_{n,j,k}\ k \in \mathscr{F}], \quad j \in \mathscr{J}, n \in \mathscr{N}_b \tag{5.28}$$

$$[V^{lo}_{1,j}, V^{up}_{1,j}] = [y_j - 2, y_j + 2\sigma], \quad j \in \mathscr{J} \tag{5.29}$$

The big-M value bounds for each data point and operator can then be calculated using the $V^{lo/up}_{n,j,k}$ with Eqs. (5.30) and (5.31). They can also be subject to a user-defined limit M_{limit} as they grow quickly with the number of layers. For division and square root, different rules for the M-values apply as their functions were formulated differently as seen in [226]. These rules are presented in Table 5.6 omitting the usually required j-indices again.

$$M^{lo}_{n,j,k} = V^{lo}_{n,j,k} - V^{up}_{n,j}, \quad n \in \mathscr{N}_b, k \in \mathscr{F} \setminus \{/, \sqrt{}\}, j \in \mathscr{J} \tag{5.30}$$

$$M^{up}_{n,j,k} = |V_{n,j}|^{lo} + V^{up}_{n,j,k}, \quad n \in \mathscr{N}_b, k \in \mathscr{F} \setminus \{/, \sqrt{}\}, j \in \mathscr{J} \tag{5.31}$$

$$M^{SC}_{n,j'} = V^{lo}_{2n,j'} - V^{up}_{2n+1,j'}, \quad n \in \mathscr{N}_b, \exists j' \in \mathscr{J} \tag{5.32}$$

The upper bounds $G^{up}_{n,j,k}$ in Eq. (5.4) are calculated applying the same logic. For the division it is simply the safety margin ρ itself and in case of the power law or square root the absolute value of the lower bound of the restricted nodal value in $g_{n,j}$ is added. For the successful application of SC, the big-M values $M^{SC}_{n,j'}$ are determined for a single data point j' deploying Eq. (5.32).

Table 5.5 Calculation of Nodal Value Bounds.

Description	Index	Lower Bound $V^{lo}_{n,j,k}$	Upper Bounds $V^{up}_{n,j,k}$
Addition	$+$	$V^{lo}_{2n}+V^{lo}_{2n+1}-\rho$	$V^{up}_{2n}+V^{up}_{2n+1}+\rho$
Subtraction	$-$	$V^{lo}_{2n}-V^{up}_{2n+1}-\rho$	$V^{up}_{2n}-V^{lo}_{2n+1}+\rho$
Multiplication	$\cdot$	$\min(arg\cdot)-\rho$	$\max(arg\cdot)+\rho$
		$arg\cdot=[V^{up}_{2n}V^{up}_{2n+1},V^{up}_{2n}V^{lo}_{2n+},V^{lo}_{2n}V^{up}_{2n+1},V^{lo}_{2n}V^{lo}_{2n+1}]$	
Division	$/$	$\min(arg/)-\rho$	$\max(arg/)+\rho$
		$arg/=\left[\frac{V^{up}_{2n}}{V^{up}_{2n+1}},\frac{V^{up}_{2n}}{V^{lo}_{2n+1}},\frac{V^{up}_{2n}}{\sqrt{\varepsilon}},\frac{V^{lo}_{2n}}{V^{up}_{2n+1}},\frac{V^{lo}_{2n}}{V^{lo}_{2n+1}},\frac{V^{lo}_{2n}}{\sqrt{\varepsilon}}\right]$	
Power Law	$\hat{}$	$-\rho$	$V^{V_{2n}}_{2n+1}+\rho$
Identity	id	$V^{lo}_{2n}-\rho$	$V^{up}_{2n}+\rho$
Exponential	e	$-\rho$	$e^{V^{up}_{2n}}+\rho$
Square Root	$\sqrt{}$	$-\rho$	$\sqrt{V^{up}_{2n}}+\rho$

Table 5.6 Calculation of big-M values for Division and Square Root.

Description	Index	Lower Bound $M^{lo}_{n,j,k}$	Upper Bounds $M^{up}_{n,j,k}$
Division	$/$	$V^{lo}_{2n}-\lvert V^{up}_{n}\rvert\max(arg)-\rho$	$V^{up}_{2n}-\lvert V^{lo}_{n}\rvert\max(arg)+\rho$
		$arg=[V^{up}_{2n+1},\lvert V^{lo}_{2n+1}\rvert]$	
Square Root	$\sqrt{}$	$V^{lo}_{2n}-\max(V^{up}_{2n},\lvert V^{lo}_{2n}\rvert)^2-\rho$	$V^{up}_{2n}+\rho$

5.2.3 Details of Modifications of the Formulation

The previously reported MINLP formulation [226] was modified in order to reduce the number of required binary variables. This is expected to advance the overall efficiency in solving the MINLP as fewer decision variables have to be determined in the global optimization. The main difference is that in the formulation used in this work the tree is always fully constructed for a given number of pre-defined tree layers NL. The predictor variables and constants can only be assigned to the lowest layer. The inclusion of an identity function allows to pass up the values without any change to a higher layer in the expression trees.

In comparison to that, in Ref. [226] predictor variables and constants are available for selection at every node in the expression tree superseding an identity function. If those leaf-node operands are selected on a higher layer, their child nodes, as well as the other subsequent lower levels, are discarded. Hence, the tree is not set up necessarily to the maximum of allowed layers. By introducing an identity function in the modified version of the MINLP, as was proposed in the theoretical formulation of the problem in the first recorded publication in the open literature on the topic in chemical engineering [225], the binaries on branch nodes for $x_{i,j}$ and ε_n can be spared. With $N = 2^{NL} - 1 = N_b + N_l$ and $N_l = 2^{(NL-1)}$, Eqs. (5.33) to (5.35) specify the number of required binary variables (B) for the herein compared MINLP formulations in order to quantify the effect in reduction and scaling behaviour.

$$B_{New} = N_b \cdot NF + N_l \cdot NX + \frac{N_l}{2} \tag{5.33}$$

$$B_{Cozad} = N - b \cdot (NF + NX + 1) + N_l \cdot (NX + 1) \tag{5.34}$$

$$\Delta B = B_{New} - B_{Cozad} = -N_b \cdot (NX + 1) - \frac{N_l}{2} \tag{5.35}$$

The delta in Eq. (5.35) proves that the significance of the reduction increases with the number of layers (*NL*) and the number of overall considered input variables (*NX*) (see SI for a quantitative comparison of scaling behaviour). Furthermore, the full construction of the tree allowed replacing big-M constraints at the leaf-nodes by equality constraints due to linearity in the binary variables. Another main difference is the asymmetric supply of continuous variables as constants. This design reduces the symmetries in the superstructure set-up and realizes a reduction in the number of required binaries as discussed above.

5.2.4 Solver

For the aims of this work, the choice of solvers is limited to those that can deterministically solve MINLPs to the global optimum. According to Ref. [228], the general list of feasible non-convex global MINLP solvers contains ANTIGONE [229], BARON [230], Couenne [231], LindoGlobal [232], and SCIP [233]. According to the results of the comparative solver study [226], BARON solves more SR problems and converges faster than all other solvers. Hence, BARON (v. 18.5.9), as a commercial general-purpose solver in deterministic global optimization, was selected in combination with IBM CPLEX [234] as a sub-solver. The optimization problem was set-up and passed on to the solver using Pyomo (v.5.5) [235]. Upon solver completion, the optimization results were analyzed within Python 3.6.5, allowing to translate the optimal decision variable values into the corresponding algebraic equation. This model can then be evaluated at different inputs for prediction as well. For further mathematical simplification of the equation, Python's library for symbolic mathematics SymPy [236] was used.

5.2.5 Physical Model Selection

The SR is to be performed with noisy experimental data. Following a globally optimal approach targeting model accuracy exclusively (Eq. (5.1)), errors are represented in the final model, what is also known as overfitting. Hence, methodological measures have to be included to restrict the influence of experiment errors on the final model and assure generalisation capabilities with low errors to unseen data. In case of SR, the errors in the training data are propagated through the expression tree and the selected operators apply to the data including errors. The limited robustness to noise is especially prevalent among SR due to its maximal flexibility in constructing free-form models [237].

To only extract the relevant terms describing the main signal and to preferably exclude the errors, the complexity of the final model is penalized. Model complexity is restricted to a threshold (C). The identity function does not add to the complexity of a model. Consequently, the true complexity must be discounted by nodes with an identity function assigned. These complexity criteria is included as an additional constraint in the MINLP (Eq. (5.36)).

$$\sum_{n\in\mathscr{N}_b}\sum_{k\in\mathscr{F}\setminus\{id\}}\delta_{n,k}+\sum_{n\in\mathscr{N}_l}\sum_{i\in\mathscr{X}}\zeta_{n,i}+\sum_{n\in\mathscr{N}_l{}^*}\kappa_n\leq C \tag{5.36}$$

By limiting the flexibility allowed, overfitting can be reduced and sparse models found. This also increases interpretability. Furthermore, this constraint filters mathematical invariants including more terms from the search space.

Next, it is proposed to identify a portfolio of the most accurate models with varying complexity (C) by solving multiple MINLPs in parallel to global optimality. In statistics, there are multiple information theoretic criterion for model selection, such as Akaike information criterion (AIC) [238], the Bayesian information criterion (BIC) [239], and others. However, the extrapolation ability is considered as one of the key advantages of the physical model. Consequently, in the method proposed one model is selected that is as sparse as possible to allow interpretation and knowledge extraction but is also as complex as necessary to describe the underlying physical system without overfitting. Hence, the portfolio of models are to be compared with regard to validation error, defined as the sum of squared differences between the predicted and the experimental values of the validation data set. Due to the growing flexibility, the training error is assumed to be the lowest for the model with the highest allowed complexity. Without requiring assumptions about the underlying true model, these can be compared quantitatively by a data set for validation to check for overfitting. With the purpose of also comparing extrapolation capabilities of the models, the validation data set is created by extracting the data points at extrema of the predictor variables. Finally, the model selection can be based on the lowest validation error which also determines the required model complexity. The framework is illustrated in Fig. 5.4.

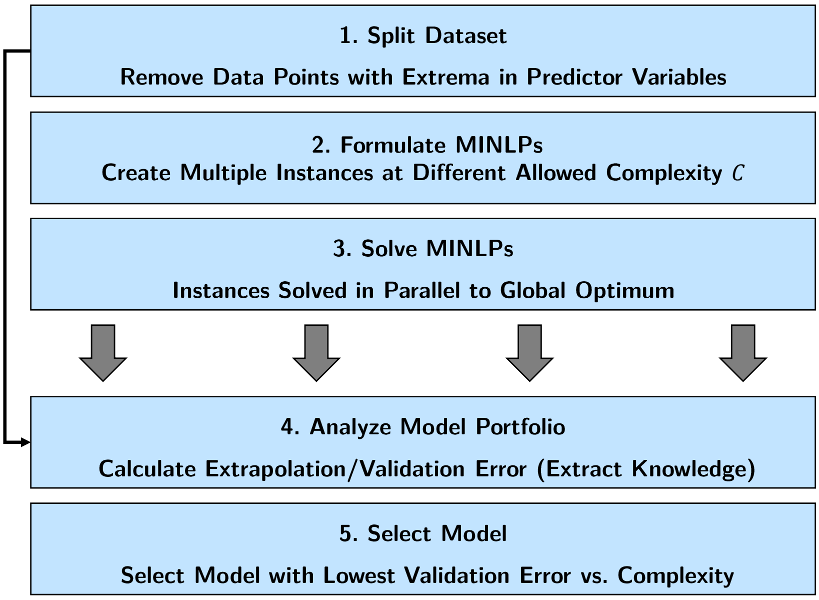

1. Split Dataset
Remove Data Points with Extrema in Predictor Variables
2. Formulate MINLPs
Create Multiple Instances at Different Allowed Complexity C
3. Solve MINLPs
Instances Solved in Parallel to Global Optimum
4. Analyze Model Portfolio
Calculate Extrapolation/Validation Error (Extract Knowledge)
5. Select Model
Select Model with Lowest Validation Error vs. Complexity

Fig. 5.4 The proposed framework for the automated identification and selection of physical models *via* symbolic regression.

5.2.6 Experimental Procedures

Automated Viscosity Measurement

Set-up of the Capillary Viscometer The experimental design for viscosity measurements follows the well-established concept of a capillary viscometer due to its simplicity and low cost [240–242]. It operates on the principles of fluidic flow in a circular tube. In general, viscosity can be determined by measuring the pressure drop at varying flow rates or vice versa.

The herein proposed experimental set-up consists of a syringe pump driving the fluid at a known flow rate through a circular tube while the pressure drop is measured. Its more detailed implementation is illustrated in Fig. 5.5 and is explained in the following. The syringe pump (CETONI neMESYS 290N) was identified as feasible pump type in order to generate constant volumetric flow rates in the range of up to 50 microliters per second ($\mu l \cdot s^{-1}$). It consists of a linear motor (M1) that drives a syringe piston in translational direction at adjustable velocities. Their advantage over other types of pumps are accurate and constant flows free from pulsation. In addition, syringe pumps have the capacity to be automated and apply elevated pressures. The selected pump has a maximum linear force of 290 Newton, which translates into a pressure of 16 bar inside the selected syringe. The syringe mounted to the pump is a CETONI glass syringe with a volume V_{syr} of 10 milliliters (ml) and an inner diameter (ID) of 14.57 millimeters (mm). The ID determines the volumetric flow rate based on piston velocity. The glass syringe was selected over plastic syringes as it withstands higher pressures. Its volume was chosen according to the small available volume of sample.

The pump is connected to an automatable three-point valve (CETONI valve, 3 bar maximum pressure) which allows to empty and refill the syringe using different routes of fluid flow. When the syringe is emptied, the valve directs the fluid through the tube while the refilling process draws fluid from a vial filled with sample. The pressure is measured at the tube inlet by a microfluidic sensor (P1, Elveflow MPS3) with a range of zero to two bar and accuracy of ± 4 millibar (mbar). This sensor was identified due to its low dead volume inside the sensor which is advantageous for small sample sizes and efficient cleaning. It was abstained from placing a second sensor at the tubing outlet to measure the actual pressure drop. A high error in pressure drop deviating from the expected value was observed in initial experiments. This is expected to be caused by the inlet effects at the second sensor. Consequently, in the present design the pressure drop is obtained relative to ambient pressure at the tube outlet, where the sample is dosed into the sample vial for reuse. The measurement of pressure drop is conducted over a stainless-steel tube of 0.5 m length (L), outer diameter of 1.59 mm and an ID of 1.02 mm which is placed horizontally at the height of the syringe to rule out influences of hydrostatic pressure. The reason for stainless steel as tube material was to ensure a straight measurement zone. The tube length L has to be sufficiently high to allow the development of a fully developed, laminar flow. Furthermore, the dimensions of ID and L were selected to balance a measurable pressure drop for low viscous fluids with limiting the pressure drop for fluids with higher viscosity. The remaining connections are realized with elastic plastic tubes with an ID of 1.5 mm.

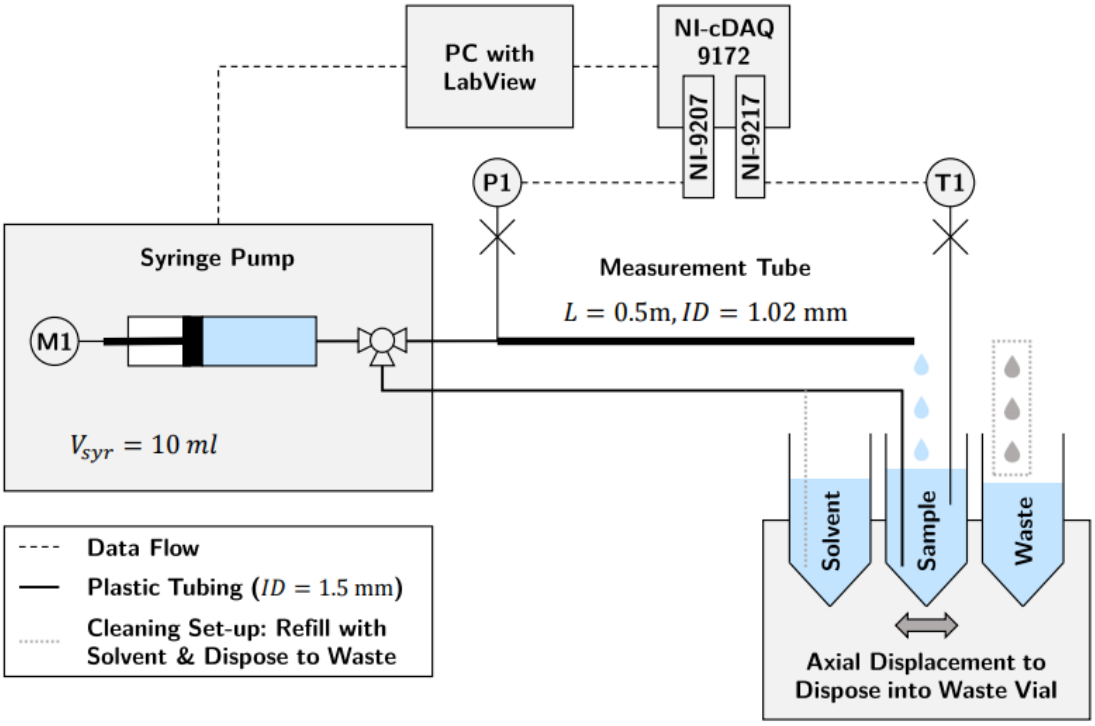

PC with LabView
NI-cDAQ 9172
NI-9207
NI-9217
P1
T1
Syringe Pump
M1
$V_{syr} = 10\,ml$
Measurement Tube
$L = 0.5\text{m}, ID = 1.02$ mm
Data Flow
Plastic Tubing ($ID = 1.5$ mm)
Cleaning Set-up: Refill with Solvent & Dispose to Waste
Solvent
Sample
Waste
Axial Displacement to Dispose into Waste Vial

Fig. 5.5 Experimental set-up of the capillary viscometer in sample measuring configuration. Solvent withdrawal and waste disposal for cleaning indicated by dashed line. Vials are displaced axially for waste disposal.

Due to temperature dependency of viscosity, each measurement is only valid at the current temperature. The set-up does not allow to control the fluid temperature to reduce set-up complexity. Nevertheless, it includes a temperature measurement (T1) by a Resistance Temperature Detector (RTD) (Bearing Sensor Platinum 100Ω, Class B, Case Style C, three-wire configuration) positioned in the sample vial. This sensor was selected over a thermocouple as it provides higher measurement accuracies at ambient conditions ($\pm 0.4\ ^{\circ}C$). Theoretically, the fluid temperature rises along the tube due to the shear stresses applied (viscous heating). For this set-up, the temperature increase was conservatively estimated to be negligible. In an energy balance of a worst-case adiabatic system, (Eq. (5.37)) the temperature increase is set equal to the viscous dissipation energy ΔE_{visc}.

$$\Delta E_{visc} = \zeta \dot{\gamma}_{avg}^2 \pi R^2 L = \dot{V} \rho c_p \Delta T \tag{5.37}$$

$$\dot{\gamma} = \frac{4\dot{V}}{\pi R^3} \tag{5.38}$$

$$\Delta T = \zeta \frac{4\dot{V}L}{\pi R^4 \rho c_p} \tag{5.39}$$

After mathematical transformations and using the expressions for the averaged shear rate from Eq. (5.38), the temperature increase can be calculated by Eq. (5.39). Under adiabatic conditions, which certainly not hold true, it would not exceed $0.4\ ^{\circ}C$. Additionally, all changes in temperature below this value could not be accurately determined due to the error range of the RTD sensor. For the estimation, the specifications of the set-up and its operational limits were used: $\dot{V}_{max} = 100\ \mu l \cdot s^{-1}$, $L_{max} = 0.5\ m$, $R_{min} = 0.5\ mm$. The fluid properties were derived by considerations of the high water content (approx. 85%) of the measured samples: $\rho_{min} = 1\ kg \cdot m^{-3}$, $\zeta_{max} = 1\ Pa \cdot s$, specific heat capacity $c_{p,min} = 3\ kJ \cdot kg^{-1}$. In conclusion of this analysis, temperature measurement and control are not required along the tube. The experimental set up in the lab is shown in Fig. 5.6.

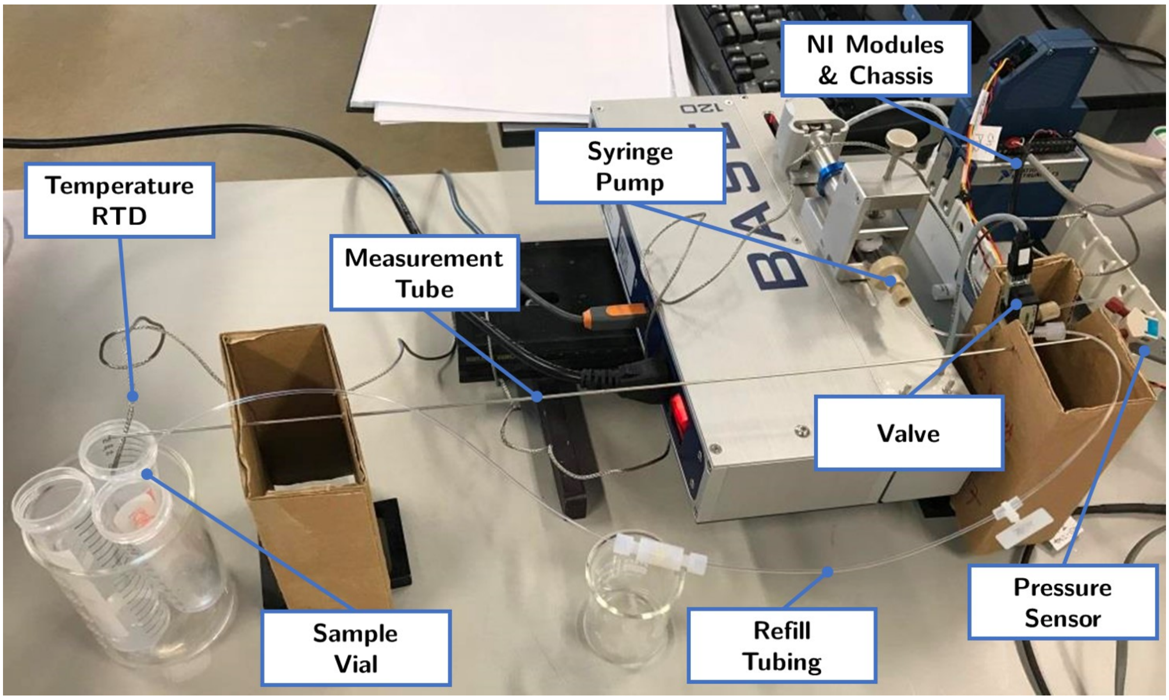


Fig. 5.6 Capillary viscometer experimental set-up assembled in lab.

Data Collection and Automation The following section provides further details about the data-collection and the automated operation of the capillary viscometer. For data-acquisition, the set-up was connected with a PC. The pressure sensor signal is acquired via a voltage and current input module (National Instruments (NI) NI-9207). Accordingly, the RTD is connected to a temperature input module (NI-9217). The modules provide functionalities such as amplifying, filtering and isolating signals. The connection with the PC is established by a chassis (NI cDAQ-9172) which serves as a power supply and plug-and-play station for multiple modules. Furthermore, to process the acquired signals as well as to manually or automatically control the syringe pump and its valve, a user interface was implemented in LabVIEW 2014 (SP1). The latter is a graphical programming language by NI. Moreover, the interface developed allows to visualize, analyze and save acquired signal values. For automated sample analysis, a special mode was established that tests multiple predefined flow rates in a sequence without any further required actions by a user. Once the syringe is empty, the valve is switched and it is refilled automatically before switching the valve again and continuing the dosing through the tube.

When dosing fluid at time t with a certain volumetric flow rate $\dot{V}_{(t)}$, a corresponding pressure drop $\Delta p_{(t)}$ can be measured. Initially, the flow develops and Δp stabilizes at a constant value. As L and R remain constant, the data of $T_{(t)}$, Δp_t and $\dot{V}_{(t)}$ are gathered for analysis. A measured time response illustrating the interaction of all variables under study is provided in Fig. 5.7. Once the syringe starts dosing ($\dot{V} \geq 0$), a pressure drop along the tube can be observed that is dependent on the level of flow rate as well as the viscosity of the fluid. With the purpose of reducing random measurement errors, the same flow rate was set at three independent runs with a dosing interval of 100 seconds. This time was selected to allow the flow to develop properly. If this time span cannot be completed due to an empty syringe, the run is repeated. This can be seen at the short peak in pressure drop before the syringe is refilled ($\dot{V} \leq 0$). Although the syringe motor stopped moving to refill the syringe, there is a long delay until the next dosing cycle begins. This setting is essential due to delayed fluid flow from vial into syringe which increases with fluid viscosity. The remaining underpressure in the syringe drives the flow until it is refilled. Hence, sufficient waiting time is required to prevent air intake through the tube outlet when switching the valve. The temperature profile can be considered as constant within one dosing run. Only minor deviations occur ($\leq 0.1°C$) that are likely to be random measurement noise.

If the analysis of one sample is finished, the tubes have to be cleaned to prevent cross-contamination with the next sample. As a solvent 2-propanol was identified due to its compatibility with the sensors and its elevated vapor pressure for the drying of the system. During the cleaning procedure, the sample vial is replaced manually by a solvent vial and the syringe is fully filled with solvent. The solvent with traces of sample is then disposed through the tube into a waste vial. In the following, the system is dried by filling the syringe with air through the tube outlet. Afterwards, the syringe is emptied using the filling route in normal operation. The two latter steps are repeated automatically until the system is dry

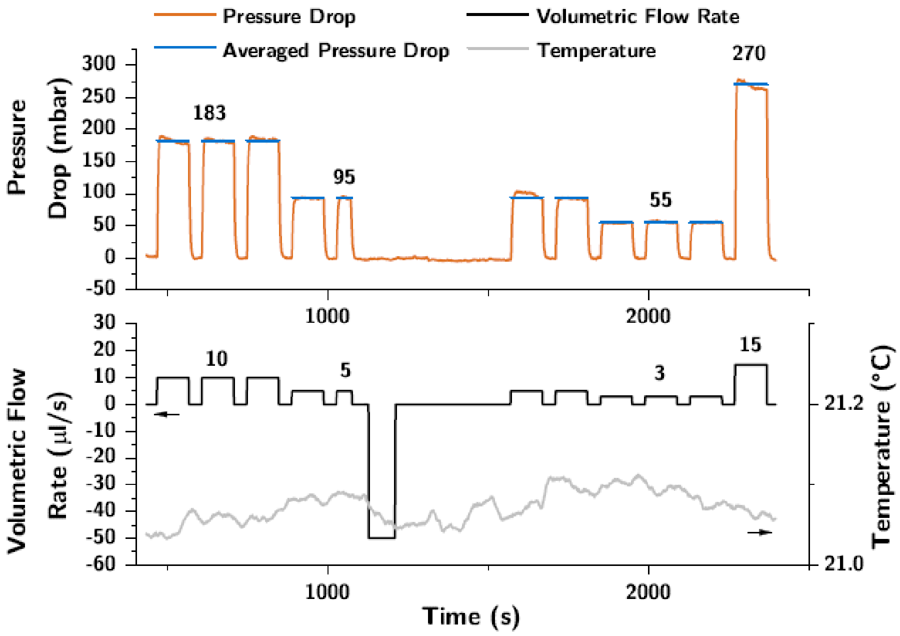
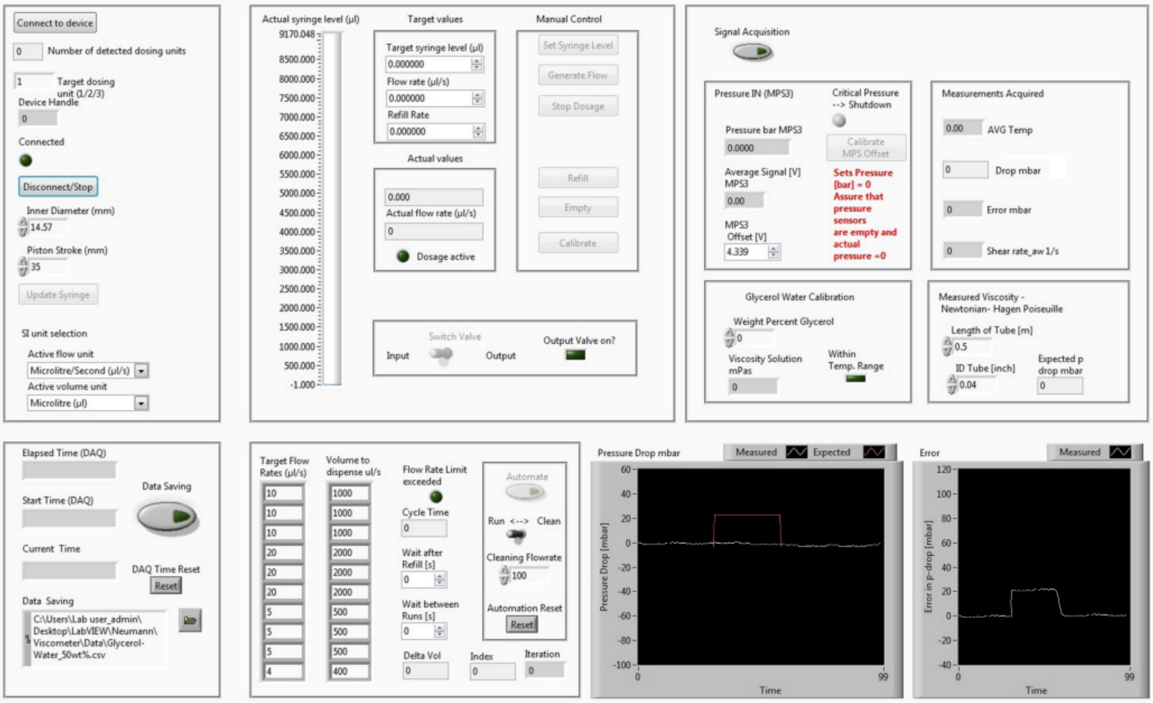


Fig. 5.7 Experimentally measured time series data of $\Delta p(t), \dot{V}_{(t)}, T_{(t)}$.

(approximately 30 minutes). For drying, moderate flow rates ($200-300\ \mu l \cdot s^{-1}$) are advised to create enough shear to break down the remaining solvent droplets in the tubing but also to allow enough time to establish a liquid-vapor equilibrium that dries the tubes.

Based on this experimental procedure, viscosity measurements of emulsions were performed. The liquid sample under study is a formulated product forming an oil-in-water (o/w) emulsion. The water phase of the emulsion is itself a mixture of water and multiple emulsifying agents (amphoteric, nonionic and anionic surfactants) at varying concentrations. Its viscosity behavior is found to be Newtonian. The oil phase consists of two polymeric oils for product thickening and conditioning at different mixing ratios and can be assumed to be a Newtonian fluid as well.

The following Fig. 5.8 shows the developed graphical user interface in LabView. It is part of the measurement and control system used for the experimental set-up for viscosity measurements. It implements all the main functions including the manual and automated control of the syringe pump, data acquisition and saving into a comma delimited file. The collected data of temperature, pressure and volumetric flow rate is acquired at a frequency of 1500 Hz with the described NI modules and is then averaged to reduce random errors. For calibration purposes, the Hagen-Poiseuille equation for Newtonian liquids and the empirical viscosity models for the viscosity of glycerol-water mixtures at varying weight contents [243, 244] were implemented. Moreover, the functionalities to measure the viscosity at different shear rates automatically and the drying procedure after the cleaning with isopropanol were designated in the interface. The actual functionalities are implemented in the backend, the block diagram.



Number of detected dosing units: 0

Target dosing unit (1/2/3): 1

Device Handle: 0

Connected

Inner Diameter (mm): 14.57

Piston Stroke (mm): 35

SI unit selection

Active flow unit: Microlitre/Second (µl/s)

Active volume unit: Microlitre (µl)

Target syringe level (µl): 0.000000

Flow rate (µl/s): 0.000000

Refill Rate: 0.000000

0.000

Actual flow rate (µl/s): 0

Pressure IN (MPS3)

Pressure bar MPS3: 0.0000

Average Signal [V] MPS3: 0.00

MPS3 Offset [V]: 4.339

Sets Pressure [bar] = 0 Assure that pressure sensors are empty and actual pressure =0

Measurements Acquired

AVG Temp: 0.00

Drop mbar: 0

Error mbar: 0

Shear rate_aw 1/s: 0

Glycerol Water Calibration

Weight Percent Glycerol: 0

Viscosity Solution mPas: 0

Measured Viscosity - Newtonian- Hagen Poiseuille

Length of Tube [m]: 0.5

ID Tube [inch]: 0.04

Expected p drop mbar: 0

Elapsed Time (DAQ):

Start Time (DAQ):

Current Time:

Data Saving: C:\Users\Lab user_admin\Desktop\LabVIEW\Neumann\Viscometer\Data\Glycerol-Water_50wt%.csv

Target Flow Rates (µl/s)	Volume to dispense ul/s
10	1000
10	1000
10	1000
20	2000
20	2000
20	2000
5	500
5	500
5	500
4	400

Cycle Time: 0

Wait after Refill [s]: 0

Wait between Runs [s]: 0

Cleaning Flowrate: 100

Delta Vol: 0

Index: 0

Iteration: 0

Fig. 5.8 LabVIEW front panel - Graphical User Interface (GUI) for the automated operation of the capillary viscometer.

Before the whole set-up was calibrated with the glycerol-water mixture, the pressure sensor (Elveflow MPS3) was calibrated in combination with the NI-9207 module. This was necessary as the supplier calibration is based on their own Elveflow amplification and data acquisition system which was not available for set-up. The sensor was calibrated with a static air pressure by closing the sensor outlet and applying a known pressure with the DRUCK digital pressure indicator. The voltage signals were then acquired in LabView for one to two minutes for each constant pressure within the interval $(0,2]$ bar with steps of 0.1 bar. Based on the arithmetic mean for each, the calibration curve was plotted in Fig. 5.9. The resulting linear fit was implemented in the LabView program, to convert the voltage signals into the pressure measured.

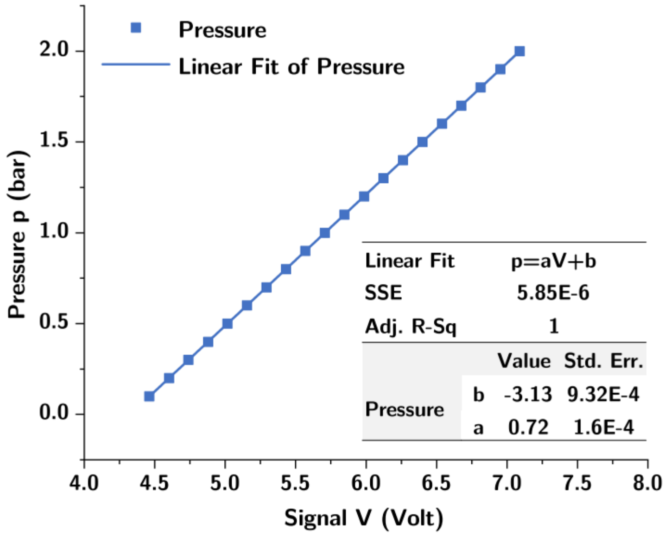


Fig. 5.9 Calibration of the pressure sensor Eleveflow MPS3 (0-2bar) with DRUCK DPI 600/IS.

Data-Processing In the case study of viscosity, the acquired time series data must be processed to derive the time-independent models of viscosity. As previously indicated in Fig. 5.10, the time series data of the pressure drop for each flow rate is averaged over time. Thereby, the information content is reduced to data triples of $\dot{V}, \Delta p$ and T at the steady-state condition which enables to calculate the corresponding shear rates and stresses (Eqs. (5.40) and (5.41)).

$$\dot{\gamma} = \frac{4\dot{V}}{\pi R^3} \tag{5.40}$$

$$\tau = \frac{\Delta p R}{2L} \tag{5.41}$$

Initially, upon dosing activation, the flow develops in the tube, for which a delayed response in pressure drop can be observed. This time interval is to be discarded in averaging. For the developed laminar flow at constant flow rate, the ideal steady-state is a constant pressure drop which is implied in the averaging scheme. A more detailed pressure profile over time and its stable part considered for averaging are depicted in Fig. 5.11. To produce the discrete data points in an automated fashion, an algorithmic averaging scheme was developed and implemented in Python.

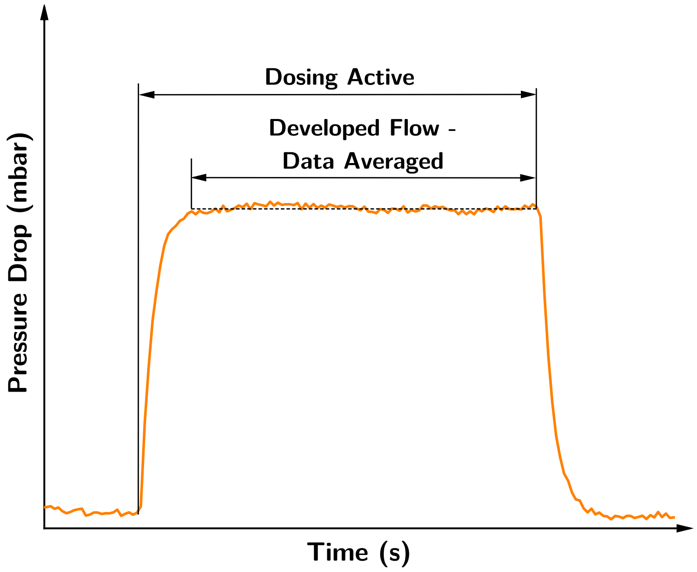


Fig. 5.10 Detailed time-dependent pressure drop profile measured with the capillary viscometer.

The procedure for data processing is summarized in Fig. 4.10. Let the set of different flow rates measured be denoted by Q with cardinality NE, which represents the final number of data points considered in the subsequent step of model building. The corresponding index is j. Data points are considered for averaging for all times t when the time series flow rate $\dot{V}_t$ is equal to Q_j. Moreover, the pressure drop at time $t, \Delta p_t$ must not deviate more than ξ from the average of K prior times series data points $\Delta p_{t-h}(h = 1, \cdots, K)$. This moving average approach ensures that only the pressure drop at steady-state condition is considered. The time interval of flow development where the pressure drop still changes significantly is

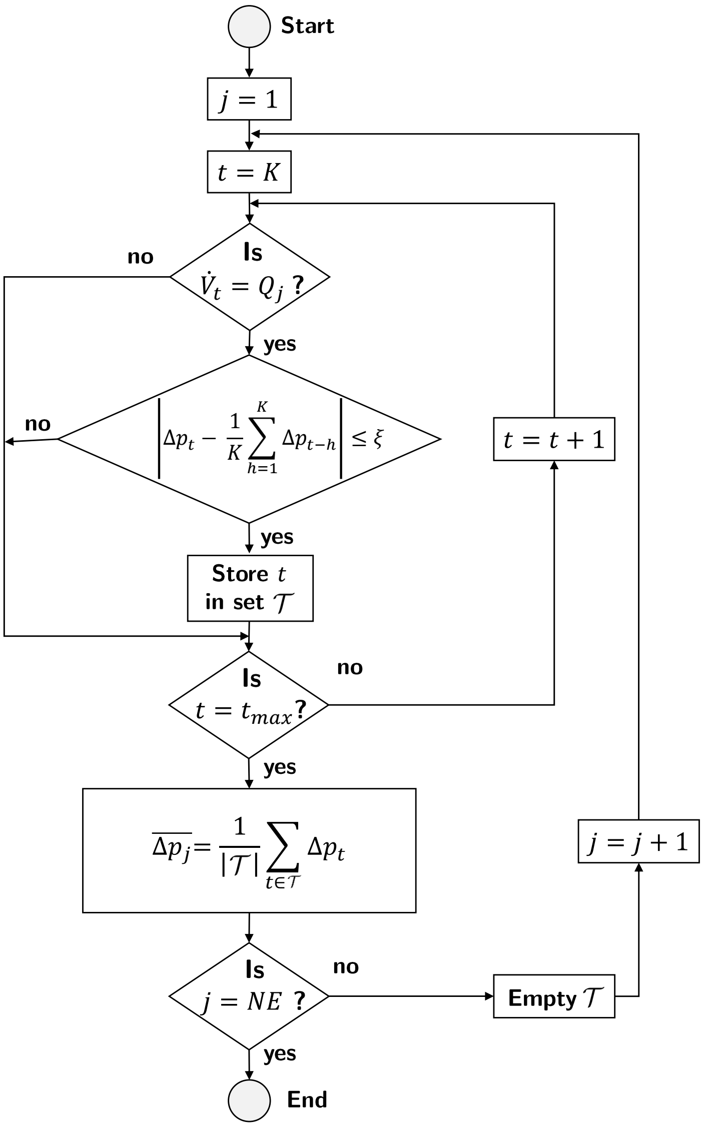

Start
$j = 1$
$t = K$
Is $\dot{V}_t = Q_j$?
no
yes
$\left| \Delta p_t - \frac{1}{K}\sum_{h=1}^{K} \Delta p_{t-h} \right| \leq \xi$
no
yes
Store t in set $\mathcal{T}$
$t = t + 1$
Is $t = t_{max}$?
no
yes
$\overline{\Delta p_j} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \Delta p_t$
$j = j + 1$
Is $j = NE$?
no
Empty $\mathcal{T}$
yes
End

Fig. 5.11 Algorithm to average time series data into discrete averaged data points at steady-state.

neglected while slowly changing trends remain in the averaged data.

Viscometer Automation and Calibration Before taking advantage of the automated set-up for measurements of the sample fluid, a calibration procedure was conducted. It aims to reduce the systematic measurement errors to enhance model identification. The calibration is important as the ideal pressure drop profile is found to be true for the results at very low flow rates only. However, it can be observed that the higher the fluid viscosity and the flow rates, the more the pressure drop decreases over time. Its beginnings are visible in the last dosing in Fig. 5.7 . This phenomenon affects the averaged pressure drop causing the measurements remaining below the true value. The potential error sources could not be removed due to missing equipment alternatives that allow automation. In conclusion, the flow rates are kept below 50 $\mu l \cdot s^{-1}$ and the measurable region of viscosities is bounded by approximately 400 $mPa \cdot s$. The quantification of this effect for the purpose of calibration as well as the potential error sources are discussed below.

At first, the pressure sensor itself was calibrated using a digital pressure indicator (Druck DPI 600/IS) applying a static pressure with air. The calibration curve can be found in the Fig. 5.9. The overall accuracy of the pressure sensor is ± 4 mbar and the corresponding reading error of the NI-9207 is $\pm 0.6\%$. The RTD for temperature measurements has an accuracy of ± 0.4 $^{\circ}C$ and an additional reading error of ± 0.2 $^{\circ}C$ introduced by the NI-9217. The quantified dosing accuracy of the syringe pump could not be identified.

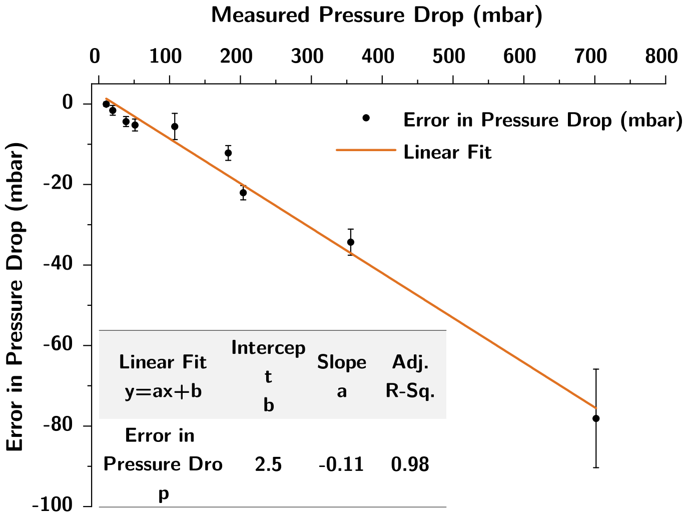
Furthermore, the full set-up was calibrated measuring the viscosities of glycerol-water mixtures at different concentrations of glycerol (60, 80 and 90 weight percentages). Due to its Newtonian behavior, the measured pressure drop and the corresponding flow rate enabled to calculate the viscosity using the HP equation. The viscosities of the same mixtures were also measured by a commercial rotational viscometer (Rheometric Scientific ARES G2) at 21.8 $^{\circ}C$. The arising temperature difference was corrected using empirical relationships for the glycerol-water mixture [243, 244]. The results of the comparison including the errors in terms of standard deviation for the capillary viscometer can be found in Table 5.7. The measurement errors of the rotational viscometer were found to be below $6E-4$ for all three mixtures and therefore are considered negligible.

Table 5.7 Water-Glycerol Mixture Calibration.

Sample (Glycerol conc.)	T	Meas. Visc.	Std. Dev.	Min.	Max	Expec. Visc.	Error
	$^{\circ}C$	$mPa\cdot s$	$mPa\cdot s$	$mPa\cdot s$	$mPa\cdot s$	$mPa\cdot s$	%
$60wt\%$	20.8	10.6	0.5	8.2	12.1	11.2	-6.0
$80wt\%$	20.1	54.4	1.2	49.0	62.3	59.1	-8.0
$90wt\%$	21.0	186.9	2.3	177.8	196.3	204.0	-8.4

As indicated earlier with non-ideal decreasing pressure drops over time, it was found that the higher the viscosity of the fluid, the more the viscosity measured by the capillary viscometer deviates negatively from the expected value. This is assumed to be a systematic error. Moreover, the standard deviation in the measurements increases notably. For quantification, a linear relationship was identified between measured pressure drop and the absolute error in pressure drop (Fig. 4.11). This could be calculated by subtracting the expected calculated pressure drop based on the rotational viscometer results. The obtained linear fit was then implemented in the data processing to correct the systematic error in future pressure drop measurements. Although the system was calibrated, the validation with both viscometers using another fluid at elevated viscosity of 161 $mPa\cdot s$ still exhibited an error of -6.5%.

Despite the usage of the equipment within the operative limits, non-constant pressure drops indicate that flow rates are not stable within one dosing process. This might be caused by small air bubbles in the syringe being compressed, a decreasing force of the pump while dosing or a leaking valve as higher pressures were needed to drive the fluids at higher viscosity. Further error sources in the overall measurement could be mistakes in the inaccurate preparation of water-glycerol mixtures, the impurity of glycerol itself or



Linear Fit y=ax+b	Intercept b	Slope a	Adj. R-Sq.
Error in Pressure Drop	2.5	-0.11	0.98

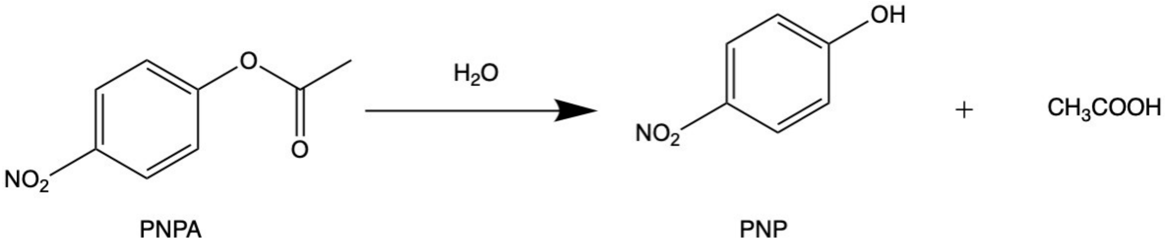
Fig. 5.12 Systematic error in pressure drop measurements of the capillary viscometer.

the cross-contamination with the cleaning solvent isopropanol due to insufficient drying. Additionally, small errors could also arise from a non-straight tube with varying diameters on a small scale, a tube orientation that is not fully horizontal as well as roughness effects inside the tube. The idealized calculations of a fully developed laminar flow neglect wall slip and potential inlet (turbulences) and outlet (dropping) effects.

In conclusion, the experimental set-up designed for automated data collection comes at the cost of elevated experimental errors as the manual data collection using commercially available rotational viscometers proves to be more accurate. Nevertheless, automated data collection is inevitable for data-driven approaches. The calibration procedure reduces the systematic error enabling the data collection at moderate accuracy in an automated fashion.

5.2.7 Reaction kinetic experiments

Reaction kinetic data were collected for hydrolysis of para nitrophenyl acetate (PNPA) under basic conditions as a case study, Scheme 5.1. For each experimental run, three stock solutions were prepared consisting of PNPA (at the desired concentration) in $3.0\%(v/v)$ aqueous methanol, 3 $mol \cdot L^{-1}$ KCl, and an aqueous NaOH solution at a fixed pH. The adopted conditions for each kinetic experiment are provided in the section below. 1 mL of each solution was directly mixed in a spectrophotometric agitated disposable cuvette. Absorption spectra (300-500 nm) were collected at fixed time intervals (Agilent, Cary 60). Absorption data at 400 nm were converted to PNP concentration. Calibration was carried out using different aqueous solutions at a known concentration of PNP at the same methanol and KCl concentrations and pH of the tested solutions. PNPA concentrations were calculated as its initial concentration minus the concentration of the formed product according to the literature [245, 246], since no by-products formation was reported under the adopted conditions.



Scheme 5.1 Hydrolysis of para-nitrophenyl acetate case study reaction scheme.

5.2.8 Materials

All chemicals (glycerol $\geq$ 99.5%, isopropyl alcohol $\geq$ 99.5%, carboxymethyl cellulose, 4-nitrophenyl acetate (esterase substrate), 4-nitrophenol $\geq$ 99.0%, potassium chloride $\geq$ 99.0%, sodium hydroxide $\geq$ 98.0%, methanol $\geq$ 99.9%) were purchased from Sigma Aldrich and used as received. Water was obtained using a Maxima (USF) Milli-Q system. Viscosity experiments on a Newtonian fluid were carried out using a commercially available emulsion.

5.3 Results and Discussion

5.3.1 Method comparison: model identification from ideal data

In the first instance, the methodology described in Section 5.2.1. was applied to data without errors to assess global optimization in SR, gaining a deeper understanding of its performance, and comparing it with parametric regression. For reasons of simplicity a function with the same structure of Arrhenius law was considered, but without units and physically relevant parameters (Eq. (5.42)). Arrhenius law is applicable in rheology as well as in reaction kinetics.

$$y = k \ \exp \frac{\eta_0}{x} \tag{5.42}$$

In our test case, $k = 3$, $\eta_0 = 8$, and $x = T$. 10 data points were randomly sampled in the x interval $(10, 40)$. Unless specified otherwise, the following calculations were performed on an Intel® Core™ i5-3337U CPU @ 1.80 GHz processor.

The parametric method ALAMO (v. 2018.4.3), introduced earlier, was selected as the benchmark method due to its sparsity promoting techniques and openly accessible user-interface. For ALAMO, linear, exponential and constant basis functions were selected as well as discrete polynomial exponents up to third order. The modelling criterion was the corrected Akaike Information Criterion (AICc) [214] . For the dataset consisting of 10 training data points, ALAMO results in a polynomial equation of third order (Table 5.8, Fig. 5.13). Even for noise-free data, only a surrogate model is found without any hints about the known underlying true relationship. This serves as an example for the limitations of parametric regression approaches in the discovery of true model structure. Nevertheless, it should be noted that the model was obtained during the first iteration in less than one second. This can be explained by the restricted search space and the high efficiency that can be achieved in solving parametric approaches.

Table 5.8 Comparison of ALAMO and symbolic regression proposed.

Approach	Identified model	Sum of Squared Errors	Comp. time (s)
Parametric Regression ALAMO	$y = 12 - 0.81x + 0.028x^2 - 0.0032x^3$	0.04	$< 1S$
Non-parametric Regression Symb. Reg. proposed	$y = \frac{e^{1}.0999}{e^{\frac{-8.0}{x}}}$	$1E-17$	29

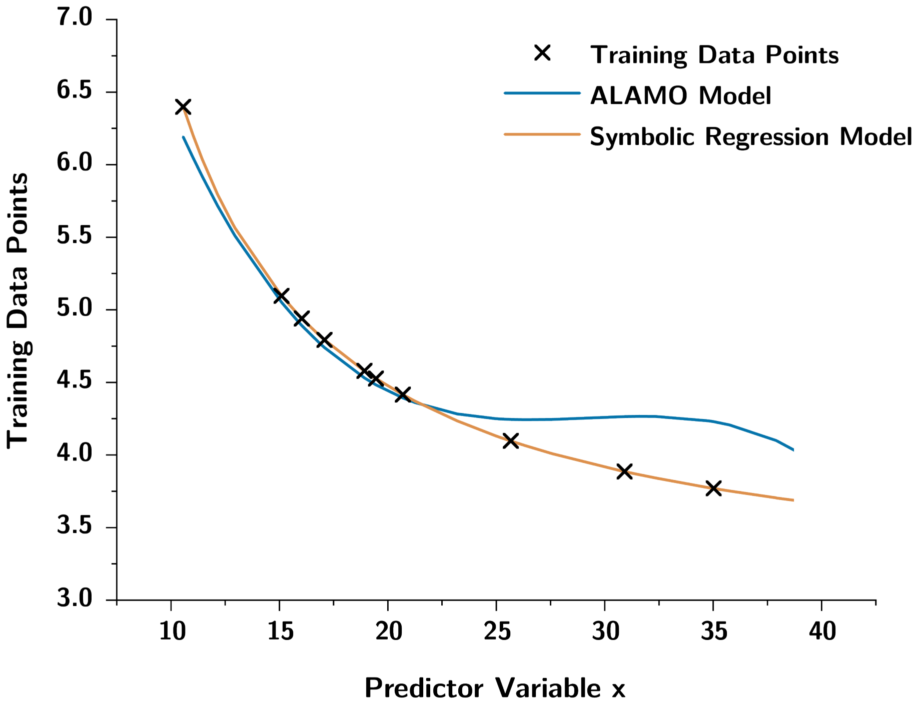


Fig. 5.13 Graphical comparison of performance of ALAMO and the proposed symbolic regression in finding a model described in Section 5.2.1.

A schematic representation of a four-layers tree for Arrhenius law identification by SR is shown in Fig. 5.14 . The resulting MINLP consisted of 54 binary and 154 continuous variables as well as 1174 constraints. The included operators were $\mathscr{F} = \{id, +, -, \cdot, /, exp\}$. The globally optimal model (Eq. (5.43)) was found within 29 s. It is a mathematically

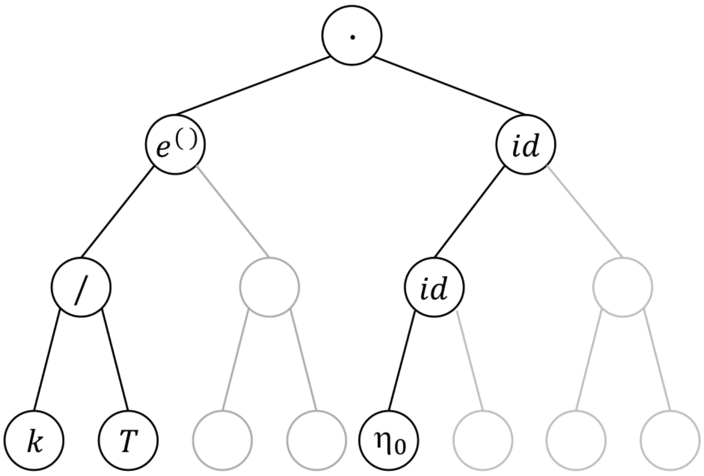


Fig. 5.14 Binary expression tree for Arrhenius equation.

invariant model of the true one as there is an unconstrained functional search space for symbolic regression.

$$y = \frac{exp(1.099)}{exp(-\frac{8.0}{x})} = 3.0\, exp\left(\frac{8.0}{x}\right) \tag{5.43}$$

In comparison to ALAMO, the SR solved to global optimality identified the true underlying model which allows significantly better generalization than the previously found surrogate model. Another advantage that can be stressed is the sparse dataset used in training. However, the structure identification without *a-priori* knowledge comes at computational cost requiring more time to converge. It is worth mentioning that the choice of the basis set of functions is crucial in both cases. However, the presented SR method is able to construct more complex functions with a limited number of basis functions.

In the second instance, the function shown in Eq. (5.44) was adapted from [226] to evaluate the impact of the modifications in the MINLP formulation (5.2.1) in terms of CPU time until convergence. The settings for the formulated MINLP problems within this section are shown in Table 5.9 The BARON solver was used with an absolute optimality gap of

$1E-5$ and a relative tolerance of 0.01. All binary variables values must satisfy an absolute integer feasibility tolerance of $1E-10$. Besides that, the default solver settings were used.

$$y = \frac{2x_1}{5-x_2} \tag{5.44}$$

Table 5.9 Settings for Evaluation and Comparisons with Ideal Data.

MINLP Parameter	$\varepsilon^{lo/up}$	V_{limit}	ρ	ε	M_{limit}
MINLP Parameter Value	$[-100,100]$	100	5	0.01	1000

In order to assess efficiency improvements with the additional symmetric cut (SC) proposed in Section 5.2.1, five different scenarios were tested based on the introduced cuts. For each, it is worked with an expression tree with tree layers ($NL=3$), 50 data points ($NE=5$) and the basic set of operators $\mathscr{F} = \{id, +, -, \cdot, /\}$. The scenarios and the corresponding CPU times are summarized in Table 5.10. Contradictorily, enabling either cut by itself, the convergence time increased compared to the case without any additional constraints. Thiscould be caused by the constraints itself that do not allow tight convex underestimations by the solver. When GC and SC are combined, the lowest convergence time could be obtained within the first four scenarios. When removing the identity function invariants ± zero from the search space and using GC and SC. Considering this inconsistency, the idea denoted in Ref. [226] of solving all scenarios in parallel and terminating once one model was found could also be applied here.

Table 5.10 MINLP Extension: Comparison in CPU Times.

		General Cut (GC) Eqs. (5.18) to (5.23)	
		Omit	Use
Symmetry Cuts (SC)	Omit	171s	432s
Eq. (5.24)	Use	338s	131s
± Zero Cut Eqs. (5.25) and (5.26)	Use GC& SC		122s

Then, in comparison with previously reported formulation [226] the scalability of globally optimal regression is evaluated. The influence of the number of included operators (NF), data points (NE), predictor variables (NX), and tree layers (NL) is studied.

The scale-up with regard to an increasing number of potential operators was studied first (Fig. 5.15). A comparison with the previously published work of [226] is not provided as, for algebraic operators, there is neither a conceptual nor a mathematical difference in the two formulations. Since the number of binary variables and big-M constraints increases linearly with the number of operators, a growth in the computational time is expected. The results obtained for $NE = 30$ and $NE = 50$ did not show a consistent linear growth in CPU time. An overall linear trend became apparent when using an increasing number of data points. The number of data points in the training set affects the number of nodal values and constraints in a linear fashion. The results obtained at different NE are summarised in Fig. 5.16 and compared with the previously reported formulation [226]. For both formulations an increase in CPU time was observed, and in all cases, our adapted formulation showed improved

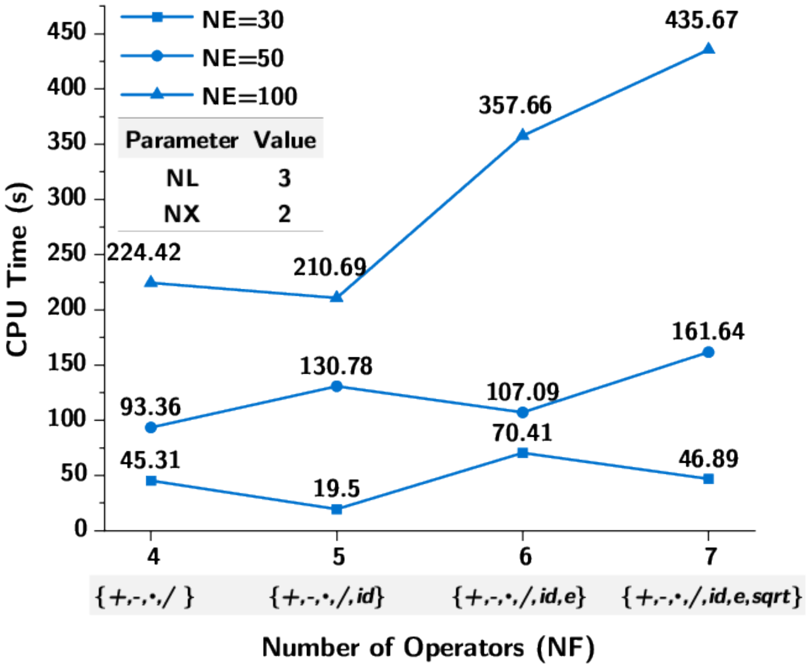


Fig. 5.15 Scalability in the number of included operators.

scalability in the number of data points. As described in Section 5.2.3, the conceptual difference in the modified formulation allows to reduce the number of binary variables which are required to allocate the predictor variables in the tree. As a result, a difference in performance should be observed when increasing the number of predictor variables. The expected improvements became evident at higher quantities of potential predictors and are shown in Fig. 5.17.

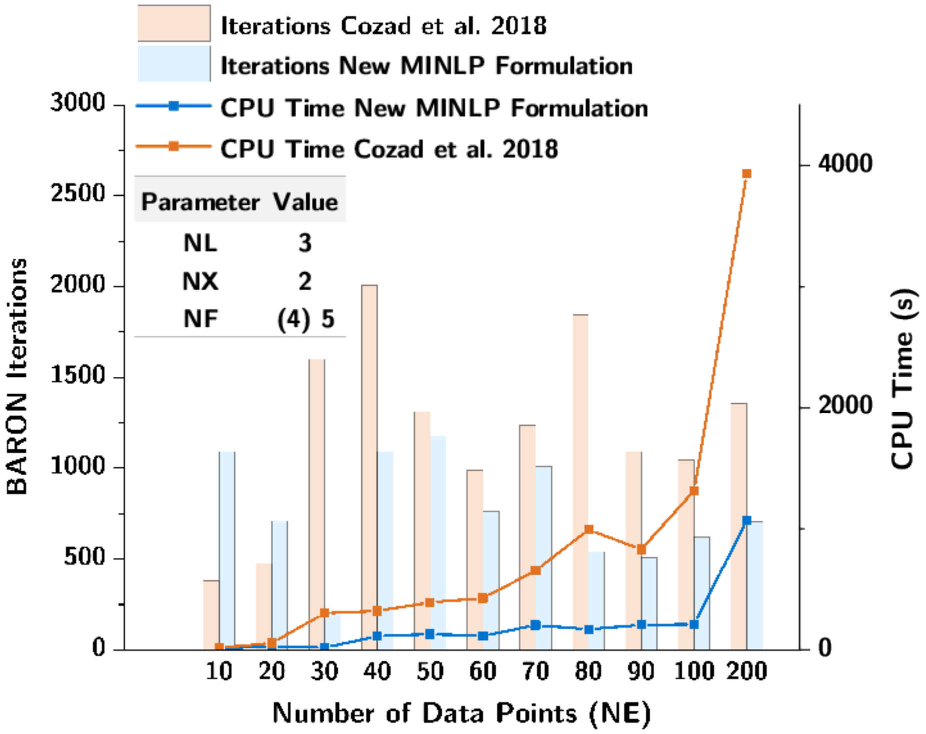


Fig. 5.16 Scalability in the number of included data points.

As the last parameter, the influence of the number of allowed layers in the expression tree was considered. By growing the tree in terms of the number of layers, an exponentially increasing number of nodes is added. Accordingly, the number of variables and constraints increases exponentially. As a result, the increase in CPU time is also exponential, as shown in Fig. 5.18. In the case of layer-scalability, the previous MINLP formulation is superior. The new proposed formulation timed out after a few hours for a number of layers greater than four. For the function under study a three-layered tree was sufficient and the ability

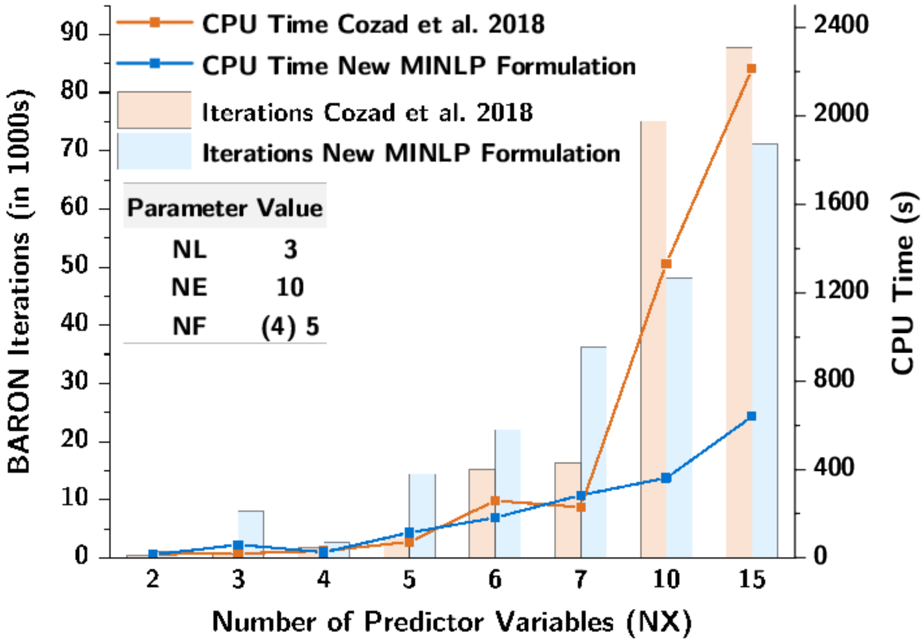


Fig. 5.17 Scalability in the number of predictor variables.

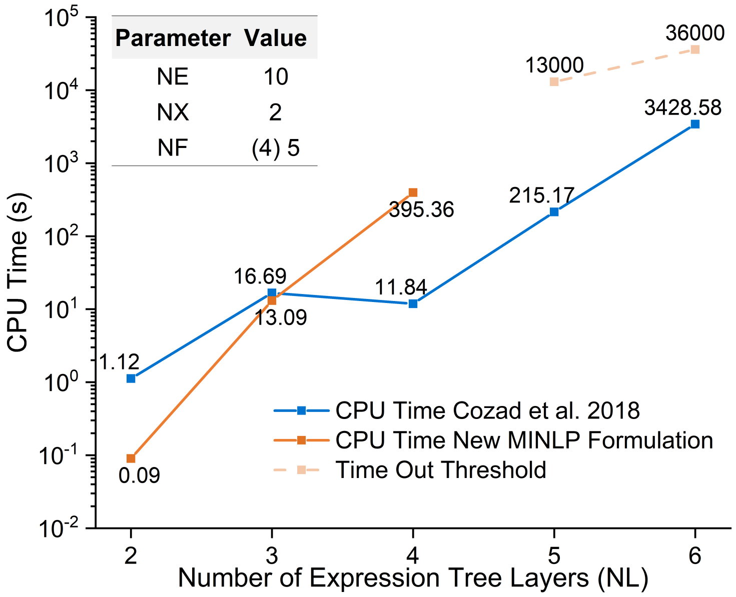


Fig. 5.18 Scalability in the number of expression tree layers.

to discard sub-layers is favoured over constructing the whole tree with identity functions. This advantage might diminish if more complex functions are sought within higher layered trees. Applicable to both formulations, this result confirms the high computational expense of SR due to the combinatorial search space. The exponential scale-up behaviour in tree layers could strongly limit the method's ability to identify more complex models. A limited investigation of the effect of the time-out time limitation for the larger trees suggests that a hyper-exponential behaviour may also be possible. A more detailed study of the scalability to more complex models is the subject of further study, which will be enabled by a significantly faster optimisation routine.

5.3.2 Physical model identification: Newton's law of viscosity

The data collected from a commercially available emulsion sample by means of the automated capillary viscometer were used to identify the simple linear relationship between shear stress (τ_w) and shear rate ($\dot{\gamma}_w$) at the wall of the tubing (5.45).

$$\tau_w = \zeta \dot{\gamma}_w \tag{5.45}$$

where

$$\tau_w = \frac{\Delta p R}{2L} \tag{5.46}$$

$$\dot{\gamma}_w = \frac{4Q}{\pi R^3} \tag{5.47}$$

For the parameter identification an expression tree with three layers, including the shear rate ($\dot{\gamma}_w$) as the only predictor variable, and ten experimental data points were used. The two extrema data points of the shear rate were not included in the training set and were used for the calculation of the extrapolation error. The set of operators included the basic operators and a power law $\mathscr{F} = \{id, +, -, \cdot, /, \wedge\}$. The resulting MINLP consisted of 28 binary and 58 continuous variables and 434 constraints. Five different instances of different complexities $C = \{3, 4, 5, 6, 7\}$ were solved in parallel.

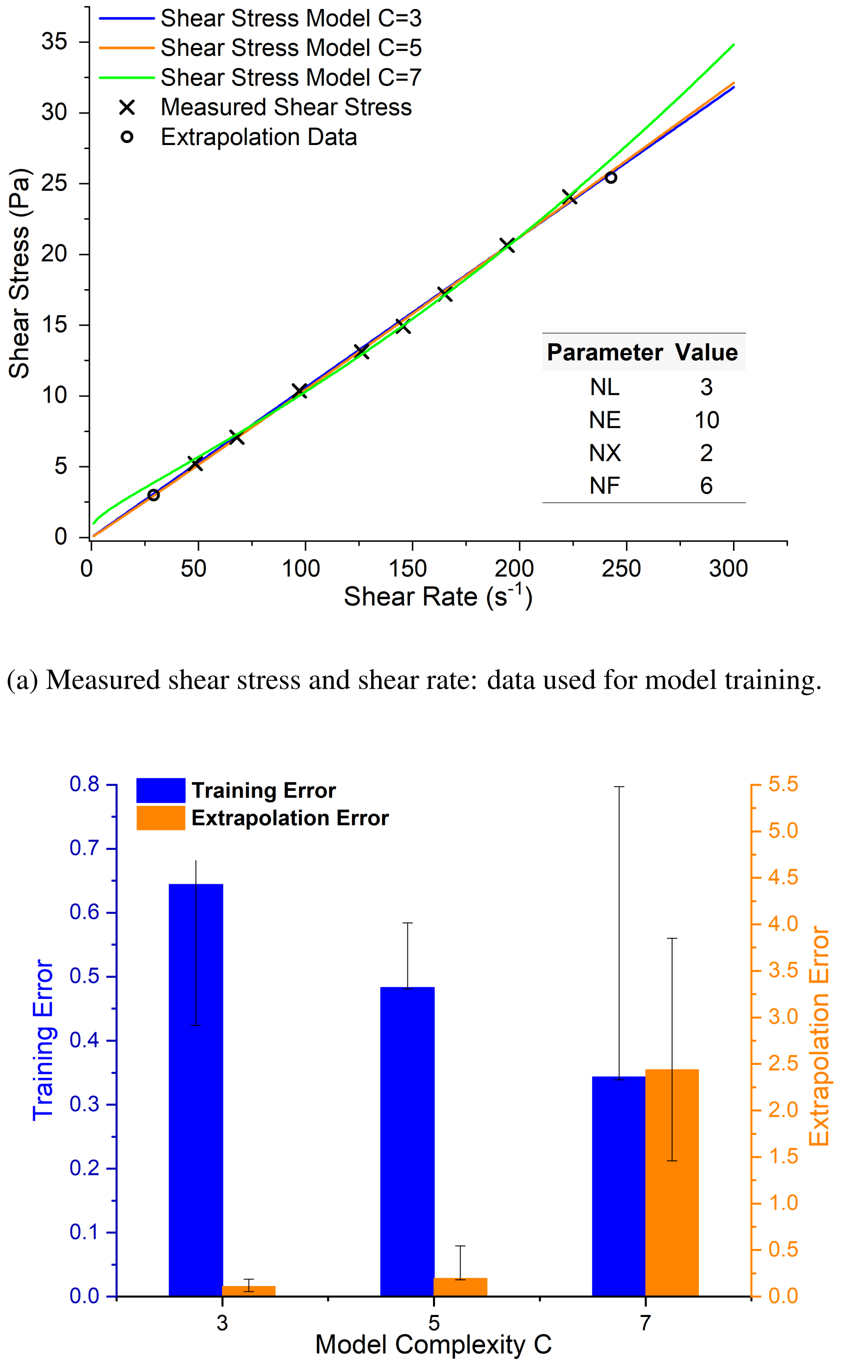
The obtained portfolio of models, Table 5.11, initially consisted of five models. As the complexity is constrained by an upper bound (inequality), similar models with the same complexity are identified. These, together with invariant models at higher complexities, were neglected. The shear rate and the shear stress were expressed as s^{-1} and *Pa*, respectively.

Table 5.11 Physical model identification: Newton's law of viscosity.

Model Complexity	Identified model	Training Error	Extrapolation Error	Computational Time (s)
3	$\tau(0.106 \pm 0.00042)\dot{\gamma}$	0.644	0.109	112
5	$\tau = ((0.099$ $\pm 0.000396)\dot{\gamma})^{(1.023 \pm 0.00102)}$	0.483	0.197	2926
7	$\tau = \dot{\gamma}^{(0.213 \pm 0.00053)\dot{\gamma}^{(0.188 \pm 0.00047)}}$	0.343	2.44	1139

To choose the best model among the identified ones, the prediction of the models was plotted together with the experimental data, see Fig. 5.19a, and the training and extrapolation errors were compared (Fig. 5.19b). In all cases, the error reported is the squared error, defined as the sum of the squared differences between the predicted and experimental values for each data set. As expected, the training error decreases with the complexity of the model as there is more flexibility allowed to SR. However, the comparison of the extrapolation errors shows the superiority of Newton's law model (C=3), whereas the other identified models suffer from overfitting. Overall, the Newton's law model can be selected as the sparsest model with the highest generalisation capability, and can be easily interpreted to generate knowledge about the physics of the system under investigation.

The model identification was conducted with only 10 data points, highlighting the sparsity



(a) Measured shear stress and shear rate: data used for model training.

(b) Errors in training and extrapolation data set for model identification.

Fig. 5.19 Physical model selection for Newtonian power law.

of required data in the presented method compared to other data-driven methods. This is especially beneficial in chemistry, where data points can be expensive to generate.

5.3.3 Non-Newtonian Power Law

Identification of a non-Newtonian power law was used to prove that the model selection framework favours higher complexity models where required. Eleven experimental data points were collected using 1% $w\,/\,w$ aqueous carboxymethyl cellulose at different flow rates. As for the Newton's law identification, an expression tree with three layers was used, including the shear rate ($\dot{\gamma}_w$) as the only predictor variable. Seven different instances of different complexities C=1, 2, 3, 4, 5, 6, 7 were solved in parallel. The resulting MINLPs have 24 binary and 65 continuous variables, and 486 constraints. The data were pre-processed scaling down the apparent shear rate (Eq. (5.45)) by a factor of 10 before training. Especially when including power law operations, this allowed to keep the variable bounds and big-M values calculated by interval arithmetic low, reducing the overall search space of the solver.

With these settings, the portfolio of three models, Table 5.12, was obtained within 13 min. The mathematically invariant and similar models were discarded. The models with complexities C = 2, C = 4, and C = 6, 7 were invariant to the models with complexities C = 1, C = 3, and C = 5, respectively. The resulting portfolio consists of three different models of different complexity.

It is noteworthy that all the computed models can be physically interpreted as the ones describing Newtonian fluids, Bingham fluids and non-Newtonian (power law) fluids. The model selection was carried out comparing their training and validation errors. Once again, two experimental data points at the edges of the investigated range of shear rates were taken aside and used to evaluate the extrapolation performance of the obtained models. The three candidate models together with their performance on the training and validation data are

Table 5.12 Physical model identification: Newton's law of viscosity.

Model Complexity	Identified model	Training Error	Extrapolation Error	Computational Time (s)
1	$\tau = \dot{\gamma}$	242.4	16.02	60
3	$\tau = (4.448 \pm 0.3425) + \dot{\gamma}$	12.19	9.30	314.4
5	$\tau = (0.712 \pm 0.0044)\dot{\gamma}^{(0.671 \pm 0.0017)}$	0.343	2.44	1139

shown in Fig. 5.20. In this case, both the training and the extrapolation errors decrease with complexity, indicating that the most complex power law is the most appropriate for the description of the experimental data.

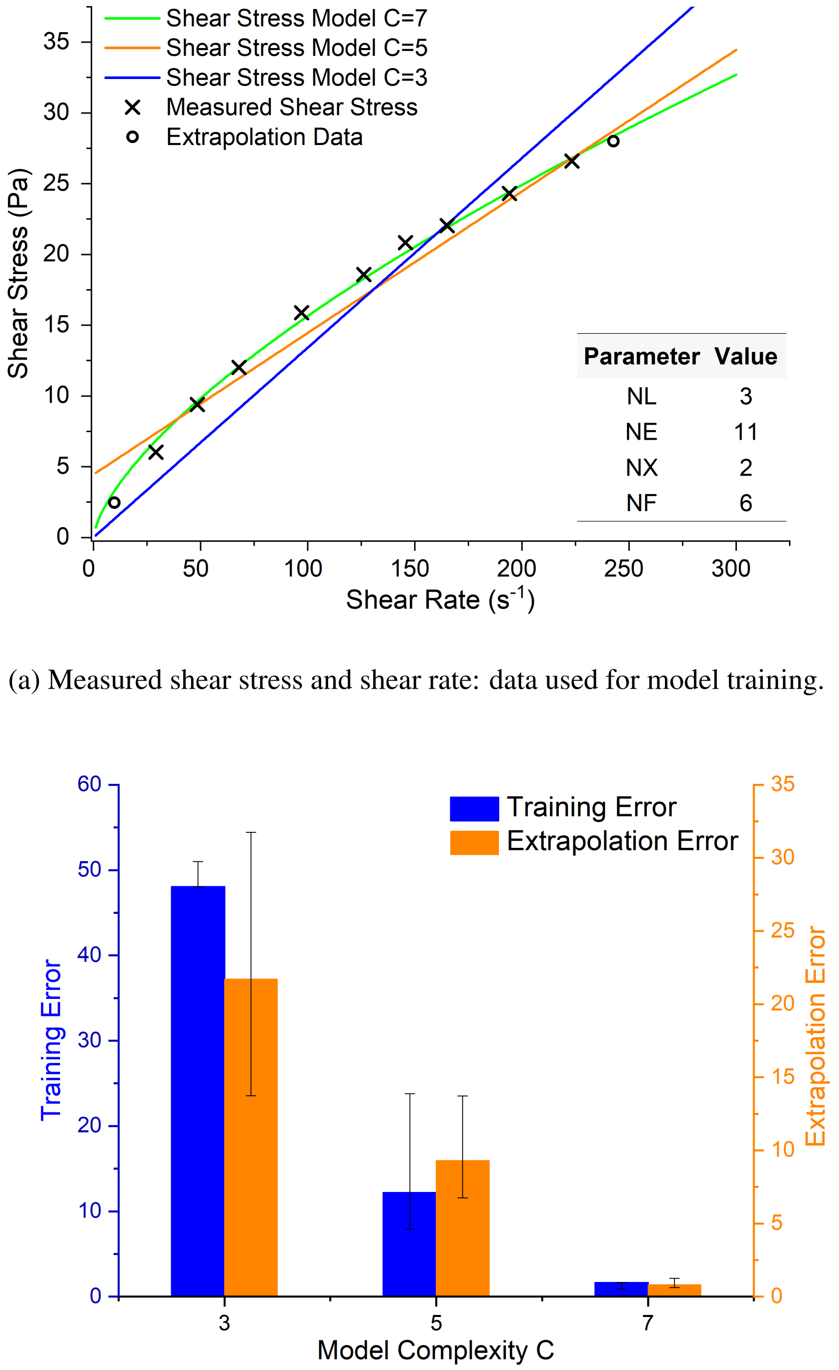
5.3.4 First-order kinetic law

Previous examples show the potential of the adopted methodology to discover sparse and interpretable models to describe the viscous behaviour of different fluids with a limited amount of experimental data. Moreover, a simple procedure was proven to be effective to select the most appropriate model within the obtained portfolio. In the following, the same procedure was applied to learn a kinetic model of a simple test reaction for which a large amount of experimental data was available.

According to Ref [53], hydrolysis of carboxylic acid esters can be described by first order kinetic law, Eq. (5.48).

$$r = k_h[PNPA] \tag{5.48}$$

where [PNPA] is the molar concentration of the investigated compound (para-nitrophenyl



(a) Measured shear stress and shear rate: data used for model training.

(b) Errors in training and extrapolation data set for model identification.

Fig. 5.20 Physical model selection for non-Newtonian power law.

acetate) and the kinetic constant k_n can be expressed as shown in Eq. (5.49)

$$k_h = k_N + k_A[H^+] + k_B[OH^-] \tag{5.49}$$

Under the adopted experimental conditions ($pH > 10.52$) the terms k_N and $k_A[H^+]$ are negligible and the overall kinetic law is given by Eq. (5.50)

$$r = k_B[OH^-][PNPA] \tag{5.50}$$

In the first attempt, experimental data were collected at a fixed pH of 10.52, at different PNPA concentrations. Concentrations vs time data series were pre-processed to obtain an approximation of the reaction rate over time using the centered difference approximation.

For this example, a three-layer tree structure was allowed, including [PNPA] as the only predictor variable. Due to the relatively low values of the measured reaction rates ($10^{-7} - 10^{-9}\ mol \cdot L^{-1} \cdot s^{-1}$), they were expressed as $mmol \cdot L^{-1} \cdot h^{-1}$.

Five MINLP instances were solved $C = \{3,4,5,6,7\}$ in parallel. The resulting MINLPs have 21 binary and 219 continuous variables and 1354 constraints. 39 experimental data points were split into 31 training examples and 8 validation data points in the range $[PNPA] \in (1.11 \cdot 10^{-3},\ 4.63 \cdot 10^{-2}\ mmol \cdot L^{-1})$. The validation points were chosen at the lower/upper end of the dataset in order to test the extrapolation ability of the model.

The model portfolio without doubling is summarized in Table 5.13, and the errors are shown in Fig. 5.21. The identified model with the lower extrapolation error is the true underlying first-order kinetic law governing the physics of the chemical system.

Table 5.13 Physical model identification: First order kinetic law.

Model Complexity	Identified model	Training Error	Extrapolation Error	Computational Time (s)
3	$r = (8.48 \pm 0.042)[PNPA]$	$3.070 \cdot 10^{-4}$	$2.365 \cdot 10^{-3}$	62.6
5	$r = (8.84 \pm 0.051)[PNPA]$ $+(0.00133 \pm 0.00109)$	$2.840 \cdot 10^{-4}$	$2.692 \cdot 10^{-3}$	84.3

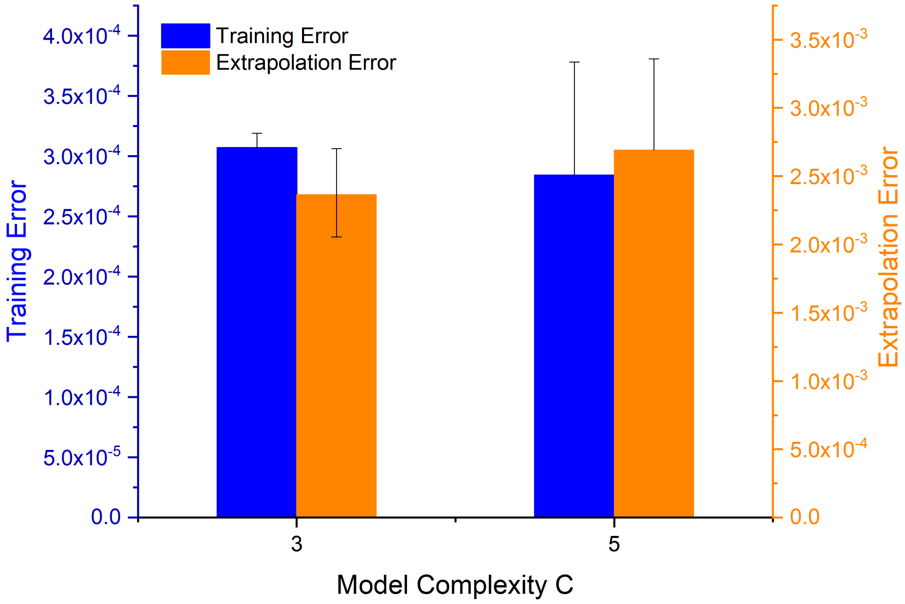


Fig. 5.21 Physical model selection for first order kinetic law identification: Errors in training and extrapolation data set for model identification.

5.3.5 First-order kinetic law: dependence on pH

In the second attempt experimental data collected at different pH were included in the training algorithm to identify the dependence of the kinetic constant on the $[OH^-]$ concentration. The training data set consisted of 80 experimental data obtained varying $[PNPA]$ and $[OH^-]$ in the ranges $5.04 \cdot 10^{-4} - 4.55 \cdot 10^{-2}$ $mmol \cdot L^{-1}$ and $3.33 \cdot 10^{-1} - 4.33$ $mmol \cdot L^{-1}$, respectively.

An expression tree with three layers was used again. The set of operators included the basic operators and a power law $\mathscr{F} = id, +, -, \cdot, /, \wedge$. The resulting MINLP consisted of 25 binary and 450 continuous variables, and 2973 constraints. Five different instances were solved in parallel of different complexities $C = \{3,4,5,6,7\}$. 80 experimental data points were split into 80% training examples and 20% validation data in the ranges $[PNPA] \in (5.00 \cdot 10^{-4},\ 4.63 \cdot 10^{-2}$ $mmol \cdot L^{-1})$ and $[OH] \in (3.33 \cdot 10^{-1},\ 4.33 \cdot 10^{-2}$ $mmol \cdot L^{-1})$. The validation points were chosen at the lower/upper end of the dataset in order to test the extrapolation ability of the model. Reaction rates were expressed as $mmol \cdot L^{-1} \cdot h^{-1}$.

According to the examples reported in Section 5.3.4 , invariant models were obtained for C = 4 and 6, and discarded. The obtained portfolio of models is summarized in Table 5.14 and errors in Fig. 5.22.

Table 5.14 Physical model identification: kinetics dependence on pH.

Model Complexity	Identified model	Training Error	Extrapolation Error	Computational Time (s)
3	$r = (33.2 \pm 1.7)[PNPA]$	6.70	4.59	241
5	$r = (26.6 \pm 0.08)[PNPA][OH^-]$	$2.99 \cdot 10^{-2}$	$1.79 \cdot 10^{-3}$	226
7	$r = ((24.7 \pm 0.074)[OH^-])$ $[PNPA][OH^-]$	$2.23 \cdot 10^{-2}$	$2.68 \cdot 10^{-3}$	4287

As shown in Fig. 5.22, both errors are of 2 or 3 orders of magnitude higher when $C = 3$,

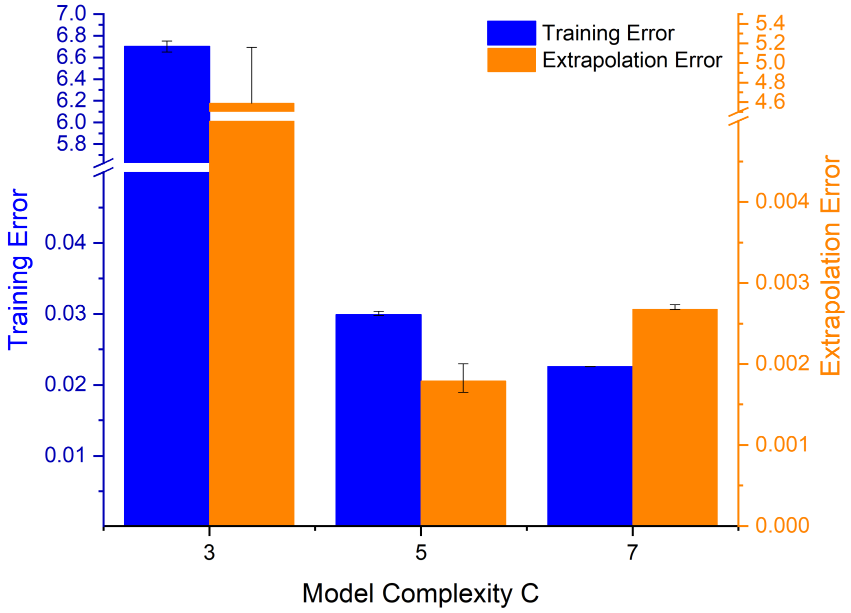


Fig. 5.22 Physical model selection for pH-dependent kinetic law identification: Errors in training and extrapolation data set for model identification.

whereas similar training errors were obtained when $C = 5$ and $C = 7$. However, the lowest extrapolation error suggests that the model with complexity $C = 5$ is the most suitable one for the description of the kinetic behaviour of the system.

5.4 Conclusions

Based on a MINLP formulation for global SR reported in the mathematical domain, a different approach in setting up the superstructure was introduced to reduce the number of binary variables involved in globally optimal SR. In addition, this formulation is complemented with a framework to enable the automated identification of physical models from crude data.

The new approach was found to outperform the previously proposed ones in terms of computational time when increasing the number of included operators, predictor variable and experimental data. As examples, the developed method allowed to correctly identify the models underlying the rheological behaviour of Newtonian and non-Newtonian fluids, as well as simple kinetic laws, also in the case of sparse data sets, which is a common scenario in chemical process development.

A significant limitation of the methodology was found in the exponential scale-up of the computational time for an increasing number of adopted layers in the tree necessary to represent complex algebraic structures of analytical function type models.

This serious issue of computational efficiency cannot be resolved by parallelization. This presently seriously limits the identification of more complex models for which a larger number of algebraic operators is often needed.

Work is currently underway to overcome these limitations using rigorous mathematical programming approaches, such as the one presented in this work, as well as complementary methodologies to derive globally optimal fitted model structures.

Chapter 6

Conclusions and Future Perspectives

6.1 Conclusions

This thesis targets the application of artificial intelligence algorithms combined with automated experimental systems in the modeling and optimization of products and process design of formulated products. Moreover, this work targets the goal of automatable, data-driven modeling techniques that derive interpretable and generalizable models for complex physical and chemical systems. The primary motivation here is not only the automated generation of a model itself but to distill knowledge about the prevalent physical phenomena from data. If the knowledge about systems that have not been understood thus far could be created automatically, scientific research could be accelerated significantly. Instead of trial-and-error based experimentation, the newly created knowledge could guide further research. Long term, the tasks of product discovery and process design could undergo a significant improvement in time and resource efficiency. Especially with regard to manufacturing scale-up, a mechanistic understanding is essential. A schematic representation of the project components and their interactions can be found in Fig. 6.1 below.

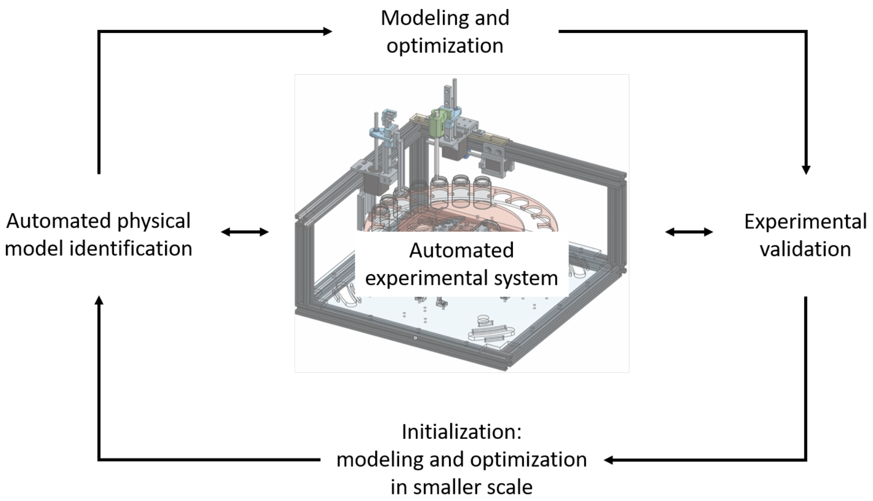

Modeling and optimization
Automated physical model identification
Automated experimental system
Experimental validation
Initialization: modeling and optimization in smaller scale

Fig. 6.1 Schematic representation of project components and their interactions.

Along the thesis there are several key insights that are worth to be mentioned again here:

- Time saving and highly reproducible robotic experiments were coupled with machine learning algorithms for the efficient optimization of the recipe of a complex formulated product of industrial interest. The optimization procedure outperformed human experts' intuition and suggested more convenient and low-priced solutions within 15 working days. The coupling of a naïve Bayes classifier with the TS-EMO algorithm allowed to take into account in the optimization procedure binary discrete outputs, also avoiding waste of time and material resources. Despite the fact that the optimization was carried out in the absence of predictive physical models, the a posteriori analysis of the Pareto front and the hyperparameters of the surrogate statistical model gave some important qualitative information about the physics of the system. Moreover, further efforts extended the method into different scenario.

- Feature engineering method as well a modified MINLP formulation for symbolic regression was proposed with the aim of deriving ohysico-chemical knowledge from noisy experimental data. As the proof of concept, key components were identified in sunscreen product design by applying feature engineering methods. Moreover, physical models of reaction kinetics and viscosity are re-discovered entirely from experimental data via symbolic regression. The purpose is to illustrate an automated research pipeline deriving interpretable and generalizable models and thereby providing access to physical knowledge.

6.2 Future perspectives

Although we have shown that artificial intelligence combined with laboratory automation can speed-up the product and process optimization process, the discovery of new phenomenon and/or the product are still challenging and not covered yet. One of the appealing potential ways is to implement the process of curious and knowledge-based inquiry inherent to human scientific research, within a reliable and high-throughput robotic system. Active searching and pooling strategies were proposed and applied in automated discovery of new chemical reactions.

Another open challenges in product design by using automated approaches is the translation of the acquired knowledge to full predictive scalability. At present, in the field of chemistry and chemical engineering, most of the data collected in lab-scale robotic platforms is used to build statistical reaction models, not taking into account scale-dependent or process-dependent interactions. At a process scale, however, mass, heat and momentum transport almost always become the most relevant controlling mechanisms, and very little information about them can be inferred from small-scale black-box optimizations. Specifically, in the field of formulated products, this can critically determine the thermodynamic state of the final product, i.e. for example its stability, shelf life, physical properties, etc. One promising way to overcome these limitations is to use robotic experiments to build generalizable physical knowledge and to learn physical models that can be integrated at a later stage to predict the behaviour of a chemical system at scale. A future challenge will be to use algorithms to find predictive correlations between the behaviours observed at the lab scale and the final properties of the processed products at the production scale. The underlying hypothesis, still to be proven or rejected, is that a direct correlation can be found between the final properties of a processed sample, e.g. stability of oil-in-water emulsion, and the observable behaviour of the two phases in a robotic platform.

The challenges of discovery and scale-up are intimately interconnected and of fundamental importance for speeding up product design and development under the constraints of the fast-changing market. Ideally, given a product of interest with certain properties, an algorithm would suggest libraries of possible combinations of ingredients and process conditions for its manufacture and rank them according to different criteria, such as practical feasibility, environmental impact, cost, etc. This may be done based on approaches similar to those currently being developed for retro- and forward synthesis in synthetic chemistry, and/or through their integration with high-throughput experimentation for discovery. And these can be the future steps of this work.

References

[1] Ka Y. Fung, Ka M. Ng, Lei Zhang, and Rafiqul Gani. A grand model for chemical product design. *Computers and Chemical Engineering*, 91:15–27, oct 2016.

[2] Rafiqul Gani and Ka M. Ng. Product design - Molecules, devices, functional products, and formulated products. *Computers and Chemical Engineering*, 81:70–79, oct 2015.

[3] Michael Hill. Chemical Product Engineering-The third paradigm. *Computers and Chemical Engineering*, 33(5):947–953, may 2009.

[4] Rafiqul Gani. Chemical product design: Challenges and opportunities. *Computers and Chemical Engineering*, 28(12):2441–2457, nov 2004.

[5] Ka M. Ng and Rafiqul Gani. Chemical product design: Advances in and proposed directions for research and teaching, jul 2019.

[6] Jonathan S. Lindsey. A retrospective on the automation of laboratory synthetic chemistry, oct 1992.

[7] Katharine Sanderson. Automation: Chemistry shoots for the Moon. *Nature*, 568(7752):577–580, apr 2019.

[8] Haralampos N. Miras, Geoffrey J.T. Cooper, De Liang Long, Hartmut Bögge, Achim Müller, Carsten Streb, and Leroy Cronin. Unveiling the transient template in the self-assembly of a molecular oxide nanowheel. *Science*, 327(5961):72–74, jan 2010.

[9] Victor Sans and Leroy Cronin. Towards dial-a-molecule by integrating continuous flow, analytics and self-optimisation, apr 2016.

[10] Artur M. Schweidtmann, Adam D. Clayton, Nicholas Holmes, Eric Bradford, Richard A. Bourne, and Alexei A. Lapkin. Machine learning meets continuous flow chemistry: Automated optimization towards the Pareto front of multiple objectives. *Chemical Engineering Journal*, 352:277–282, nov 2018.

[11] Connor W. Coley, Dale A. Thomas, Justin A.M. Lummiss, Jonathan N. Jaworski, Christopher P. Breen, Victor Schultz, Travis Hart, Joshua S. Fishman, Luke Rogers, Hanyu Gao, Robert W. Hicklin, Pieter P. Plehiers, Joshua Byington, John S. Piotti, William H. Green, A. John Hart, Timothy F. Jamison, and Klavs F. Jensen. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science*, 365(6453), aug 2019.

[12] Connor W. Coley, Regina Barzilay, Tommi S. Jaakkola, William H. Green, and Klavs F. Jensen. Prediction of Organic Reaction Outcomes Using Machine Learning. *ACS Central Science*, 3(5):434–443, may 2017.

[13] Andrew McNally, Benjamin Haffemayer, Beatrice SL Collins, and Matthew J Gaunt. Palladium-catalysed c–h activation of aliphatic amines to give strained nitrogen heterocycles. *Nature*, 510(7503):129–133, 2014.

[14] Claudia Houben, Nicolai Peremezhney, Alexandr Zubov, Juraj Kosek, and Alexei A. Lapkin. Closed-Loop Multitarget Optimization for Discovery of New Emulsion Polymerization Recipes. *Organic Process Research and Development*, 19(8):1049–1053, mar 2015.

[15] James M. Clomburg, Anna M. Crumbley, and Ramon Gonzalez. Industrial biomanufacturing: The future of chemical production, jan 2017.

[16] Jason F. Patrick, Maxwell J. Robb, Nancy R. Sottos, Jeffrey S. Moore, and Scott R. White. Polymers with autonomous life-cycle control, dec 2016.

[17] K.M. Ng, R. Gani, and K. Dam-Johansen. *Chemical Product Design: Towards a Perspective through Case Studies*, volume 23. Elsevier, 2006.

[18] Gerard Cummins and Marc P.Y. Desmulliez. Inkjet printing of conductive materials: A review, 2012.

[19] Sze Kee Tam, Ka Yip Fung, Grace Sum Hang Poon, and Ka Ming Ng. Product design: Metal nanoparticle-based conductive inkjet inks. *AIChE Journal*, 62(8):2740–2753, aug 2016.

[20] Senthil K Chandrasegaran, Karthik Ramani, Ram D Sriram, Imré Horváth, Alain Bernard, Ramy F Harik, and Wei Gao. The evolution, challenges, and future of knowledge representation in product design systems. *Computer-aided design*, 45(2):204–228, 2013.

[21] Heiko Struebing, Zara Ganase, Panagiotis G. Karamertzanis, Eirini Siougkrou, Peter Haycock, Patrick M. Piccione, Alan Armstrong, Amparo Galindo, and Claire S. Adjiman. Computer-aided molecular design of solvents for accelerated reaction kinetics. *Nature Chemistry*, 5(11):952–957, nov 2013.

[22] Michele Mattei, Elisa Conte, Georgios M Kontogeorgis, and Rafiqul Gani. Prediction of thermo-physical properties of liquid formulated products. In *Product Design and Engineering: Formulation of Gels and Pastes*. Wiley Online Library, 2013.

[23] Nick D. Austin, Nikolaos V. Sahinidis, and Daniel W. Trahan. Computer-aided molecular design: An introduction and review of tools, applications, and solution techniques. *Chemical Engineering Research and Design*, 116:2–26, dec 2016.

[24] Lik Yin Ng, Nishanth G Chemmangattuvalappil, and Denny KS Ng. Robust chemical product design via fuzzy optimisation approach. *Computers & Chemical Engineering*, 83:186–202, 2015.

[25] Nikolaos V. Sahinidis, Mohit Tawarmalani, and Minrui Yu. Design of alternative refrigerants via global optimization. *AIChE Journal*, 49(7):1761–1775, jul 2003.

[26] Lei Zhang, Stefano Cignitti, and Rafiqul Gani. Generic mathematical programming formulation and solution for computer-aided molecular design. *Computers and Chemical Engineering*, 78:79–84, jul 2015.

[27] Luke Achenie, Venkat Venkatasubramanian, and Rafiqul Gani. *Computer aided molecular design: theory and practice*. Elsevier, 2002.

[28] Nick D. Austin, Nikolaos V. Sahinidis, and Daniel W. Trahan. Computer-aided molecular design: An introduction and review of tools, applications, and solution techniques. *Chemical Engineering Research and Design*, 116:2–26, dec 2016.

[29] Nick D. Austin, Nikolaos V. Sahinidis, Ivan A. Konstantinov, and Daniel W. Trahan. COSMO-based computer-aided molecular/mixture design: A focus on reaction solvents. *AIChE Journal*, 64(1):104–122, jan 2018.

[30] J. L. Franklin. Prediction of Heat and Free Energies of Organic Compounds. *Industrial and Engineering Chemistry*, 41(5):1070–1076, may 1949.

[31] Amol Shivajirao Hukkerikar, Bent Sarup, Antoon Ten Kate, Jens Abildskov, Gürkan Sin, and Rafiqul Gani. Group-contribution + (GC +) based estimation of properties of pure components: Improved property estimation and uncertainty analysis. *Fluid Phase Equilibria*, 321:25–43, may 2012.

[32] Aage Fredenslund, Russell L. Jones, and John M. Prausnitz. Group-contribution estimation of activity coefficients in nonideal liquid mixtures. *AIChE Journal*, 21(6):1086–1099, nov 1975.

[33] R Gani and E A Brignole+. Molecular design of solvents for liquid extraction based on UNIFAC. *Fluid Phase Equilibria*, 13:331–340, 1983.

[34] W G Chapmanls2, K E Gubbinsl, G Jacksonl, and M Radosz2. SAFT: Equation-of-State Solution Model for Associating Fluids. *Fluid Phase Equilibra*, 52:31–38, 1989.

[35] Nick D. Austin, Nikolaos V. Sahinidis, and Daniel W. Trahan. A COSMO-based approach to computer-aided mixture design. *Chemical Engineering Science*, 159:93–105, feb 2017.

[36] Nick D. Austin, Apurva P. Samudra, Nikolaos V. Sahinidis, and Daniel W. Trahan. Mixture design using derivative-free optimization in the space of individual component properties. *AIChE Journal*, 62(5):1514–1530, may 2016.

[37] Christoph Gertig, Kai Leonhard, and André Bardow. Computer-aided molecular and processes design based on quantum chemistry: current status and future prospects, mar 2020.

[38] Lei Zhang, Haitao Mao, Qilei Liu, and Rafiqul Gani. Chemical product design – recent advances and perspectives, mar 2020.

[39] Fernando P Bernardo and Pedro M Saraiva. A conceptual model for chemical product design. *AIChE Journal*, 61(3):802–815, 2015.

[40] Georgios M. Kontogeorgis, Michele Mattei, Ka M. Ng, and Rafiqul Gani. An Integrated Approach for the Design of Emulsified Products. *AIChE Journal*, 65(1):75–86, jan 2019.

[41] Rafiqul Gani and Ka M Ng. Product design–molecules, devices, functional products, and formulated products. *Computers & Chemical Engineering*, 81:70–79, 2015.

[42] W.D. Seider, D.R. Lewin, J.D. Seader, S. Widagdo, R. Gani, and K.M. Ng. *Product and Process Design Principles, Synthesis, Analysis and Evaluation*. Wiley, fourth edition, 2017.

[43] Lei Zhang, Ka Yip Fung, Xiang Zhang, Ho Ki Fung, and Ka Ming Ng. An integrated framework for designing formulated products. *Computers and Chemical Engineering*, 107:61–76, dec 2017.

[44] Lei Zhang, Ka Yip Fung, Christianto Wibowo, and Rafiqul Gani. Advances in chemical product design. *Reviews in Chemical Engineering*, 34(3):319–340, apr 2018.

[45] Mohammed I. Jeraal, Nicholas Holmes, Geoffrey R. Akien, and Richard A. Bourne. Enhanced process development using automated continuous reactors by self-optimisation algorithms and statistical empirical modelling. *Tetrahedron*, 74(25):3158–3164, jun 2018.

[46] Carlos Mateos, María José Nieves-Remacha, and Juan A. Rincón. Automated platforms for reaction self-optimization in flow, sep 2019.

[47] Claudia Houben and Alexei A. Lapkin. Automatic discovery and optimization of chemical processes, jul 2015.

[48] Christopher S. Horbaczewskyj, Charlotte E. Willans, Alexei A. Lapkin, and Richard A. Bourne. An introduction to Closed-Loop Process Optimization and Online Analysis. In Alexei A. Lapkin, editor, *Green Chemical Engineering*, volume 12, chapter 12, pages 329–369. Wiley-VCH Verlag GmbH & Co. KGaA, Weinheim, Germany, 1 edition, 2018.

[49] Tanja Dimitrov, Christoph Kreisbeck, Jill S. Becker, Alán Aspuru-Guzik, and Semion K. Saikin. Autonomous Molecular Design: Then and Now, mar 2019.

[50] Alon B. Henson, Piotr S. Gromski, and Leroy Cronin. Designing Algorithms to Aid Discovery by Chemical Robots. *ACS Central Science*, 4(7):793–804, jul 2018.

[51] Anne Catherine Bédard, Andrea Adamo, Kosi C. Aroh, M. Grace Russell, Aaron A. Bedermann, Jeremy Torosian, Brian Yue, Klavs F. Jensen, and Timothy F. Jamison. Reconfigurable system for automated optimization of diverse chemical reactions. *Science*, 361(6408):1220–1225, sep 2018.

[52] Sebastian Steiner, Jakob Wolf, Stefan Glatzel, Anna Andreou, Jarosław M. Granda, Graham Keenan, Trevor Hinkley, Gerardo Aragon-Camarasa, Philip J. Kitson, Davide Angelone, and Leroy Cronin. Organic synthesis in a modular robotic system driven by a chemical programming language. *Science*, 363(6423), jan 2019.

[53] Laurie J. Points, James Ward Taylor, Jonathan Grizou, Kevin Donkers, and Leroy Cronin. Artificial intelligence exploration of unstable protocells leads to predictable properties and discovery of collective behavior. *Proceedings of the National Academy of Sciences of the United States of America*, 115(5):885–890, jan 2018.

[54] Dario Caramelli, Daniel Salley, Alon Henson, Gerardo Aragon Camarasa, Salah Sharabi, Graham Keenan, and Leroy Cronin. Networking chemical robots for reaction multitasking. *Nature Communications*, 9(1):1–10, dec 2018.

[55] Vincenza Dragone, Victor Sans, Alon B. Henson, Jaroslaw M. Granda, and Leroy Cronin. An autonomous organic reaction search engine for chemical reactivity. *Nature Communications*, 8(1):1–8, aug 2017.

[56] Alon Henson, Juan Manuel Parrilla Gutierrez, Trevor Hinkley, Soichiro Tsuda, and Leroy Cronin. Towards heterotic computing with droplets in a fully automated droplet-maker platform. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 373(2046):20140221, jul 2015.

[57] Daniel Salley, Graham Keenan, Jonathan Grizou, Abhishek Sharma, Sergio Martín, and Leroy Cronin. A nanomaterials discovery robot for the Darwinian evolution of shape programmable gold nanoparticles. *Nature Communications*, 11(1):1–7, dec 2020.

[58] Philip J Kitson, Stefan Glatzel, and Leroy Cronin. The digital code driven autonomous synthesis of ibuprofen automated in a 3D-printer-based robot. *Beilstein Journal of Organic Chemistry*, 12(1):2776–2783, dec 2016.

[59] Muhammad E. Abdelhamid, Timothy Murdoch, Tamar L. Greaves, Anthony P. O'Mullane, and Graeme A. Snook. High-throughput approach for the identification of anilinium-based ionic liquids that are suitable for electropolymerisation. *Physical Chemistry Chemical Physics*, 17(27):17967–17972, jul 2015.

[60] Julie Anthoni, Latifa Chebil, Frederic Lionneton, Jacques Magdalou, Catherine Humeau, and Mohamed Ghoul. Automated analysis of synthesized oligorutin and oligoesculin by laccase. *Canadian Journal of Chemistry*, 89(8):964–970, aug 2011.

[61] Hans-Peter Buchstaller and Uwe Anlauf. Parallel Solution-Phase Synthesis of a 2-Aminothiazole Library Including Fully Automated Work-Up. *Combinatorial Chemistry & High Throughput Screening*, 14(2):104–108, jan 2011.

[62] Elina Buitrago, Helena Lundberg, Hans Andersson, Per Ryberg, and Hans Adolfsson. High Throughput Screening of a Catalyst Library for the Asymmetric Transfer Hydrogenation of Heteroaromatic Ketones: Formal Syntheses of (R)-Fluoxetine and (S)-Duloxetine. *ChemCatChem*, 4(12):2082–2089, dec 2012.

[63] Melodie Christensen, Folarin Adedeji, Shane Grosser, Kerstin Zawatzky, Yining Ji, Jinchu Liu, Jon A. Jurica, John R. Naber, and Jason E. Hein. Development of an automated kinetic profiling system with online HPLC for reaction optimization. *Reaction Chemistry and Engineering*, 4(9):1555–1558, sep 2019.

[64] Amy A. Cockram, Robert D. Bradley, Sylvie A. Lynch, Patricia C.D. Fleming, Neal S.J. Williams, Martin W. Murray, Simon N. Emmett, and Steven P. Armes. Optimization of the high-throughput synthesis of multiblock copolymer nanoparticles in aqueous media: Via polymerization-induced self-assembly. *Reaction Chemistry and Engineering*, 3(5):645–657, oct 2018.

[65] Peng Cui, David P. Mcmahon, Peter R. Spackman, Ben M. Alston, Marc A. Little, Graeme M. Day, and Andrew I. Cooper. Mining predicted crystal structure landscapes with high throughput crystallisation: Old molecules, new insights. *Chemical Science*, 10(43):9988–9997, nov 2019.

[66] Erik Fenster, Toby R. Long, Qin Zang, David Hill, Benjamin Neuenswander, Gerald H. Lushington, Aihua Zhou, Conrad Santini, and Paul R. Hanson. Automated synthesis of a 184-member library of thiadiazepan-1,1-dioxide-4- ones. *ACS Combinatorial Science*, 13(3):244–250, may 2011.

[67] Martin W. M. Fijten, Michael A. R. Meier, Richard Hoogenboom, and Ulrich S. Schubert. Automated parallel investigations/optimizations of the reversible addition-fragmentation chain transfer polymerization of methyl methacrylate. *Journal of Polymer Science Part A: Polymer Chemistry*, 42(22):5775–5783, nov 2004.

[68] David Fournier, Richard Hoogenboom, Hanneke M.L. Thijs, Renzo M. Paulus, and Ulrich S. Schubert. Tunable pH- and temperature-sensitive copolymer libraries by reversible addition-fragmentation chain transfer copolymerizations of methacrylates. *Macromolecules*, 40(4):915–920, feb 2007.

[69] R. L. Greenaway, V. Santolini, M. J. Bennison, B. M. Alston, C. J. Pugh, M. A. Little, M. Miklitz, E. G.B. Eden-Rump, R. Clowes, A. Shakil, H. J. Cuthbertson, H. Armstrong, M. E. Briggs, K. E. Jelfs, and A. I. Cooper. High-throughput discovery of organic cages and catenanes using computational screening fused with robotic synthesis. *Nature Communications*, 9(1):1–11, dec 2018.

[70] Ibtissem Jlalia, Claire Beauvineau, Sophie Beauvière, Esra Önen, Marie Aufort, Aymeric Beauvineau, Eihab Khaba, Jean Herscovici, Faouzi Meganem, and Christian Girard. Automated Synthesis of a 96 Product-Sized Library of Triazole Derivatives Using a Solid Phase Supported Copper Catalyst. *Molecules*, 15(5):3087–3120, apr 2010.

[71] Rachel J. Kearsey, Ben M. Alston, Michael E. Briggs, Rebecca L. Greenaway, and Andrew I. Cooper. Accelerated robotic discovery of type II porous liquids. *Chemical Science*, 10(41):9454–9465, oct 2019.

[72] Anna M. Maj, Svetlana Heyte, Marcia Araque, Franck Dumeignil, Sébastien Paul, Isabelle Suisse, and Francine Agbossou-Niedercorn. First catalytic asymmetric hydrogenation of quinoxaline-2-carboxylates. *Tetrahedron*, 72(10):1375–1380, mar 2016.

[73] László I. Majoros, Bernard Dekeyser, Richard Hoogenboom, Martin W. M. Fijten, Nancy Haucourt, and Ulrich S. Schubert. Solution prepolymerization as a new route for preparing aliphatic polyurethane prepolymers using high-throughput experimentation. *Journal of Polymer Science Part A: Polymer Chemistry*, 47(15):3729–3739, aug 2009.

[74] Mohammed J. Nasrullah, James A. Bahr, Christy Gallagher-Lein, Dean C. Webster, Richard R. Roesler, and Peter Schmitt. Automated parallel polyurethane dispersion synthesis and characterization. *Journal of Coatings Technology and Research*, 6(1):1–10, 2009.

[75] Norbert Stoll, Arne Allwardt, Uwe Dingerdissen, and Kerstin Thurow. An 8-fold parallel reactor system for combinatorial catalysis research. *Journal of Automated Methods and Management in Chemistry*, 2006, 2006.

[76] Nicholas M.K. Tse, Danielle F. Kennedy, Bradford A. Moffat, Nigel Kirby, Rachel A. Caruso, and Calum J. Drummond. High-throughput preparation of hexagonally ordered mesoporous silica and gadolinosilicate nanoparticles for use as MRI contrast agents. *ACS Combinatorial Science*, 14(8):443–450, aug 2012.

[77] Gisbert Schneider. Automating drug discovery, feb 2018.

[78] Michael Shevlin. Practical High-Throughput Experimentation for Chemists. *ACS Medicinal Chemistry Letters*, 8(6):601–607, jun 2017.

[79] Hui Zhao, Olivier Graf, Nebojsa Milovic, Xiaosong Luan, Markus Bluemel, Markus Smolny, and Kurt Forrer. Formulation development of antibodies using robotic system and High-Throughput Laboratory (HTL). *Journal of Pharmaceutical Sciences*, 99(5):2279–2294, may 2010.

[80] Andreas Håkansson. Rotor-Stator Mixers: From Batch to Continuous Mode of Operation—A Review. *Processes*, 6(4):32, apr 2018.

[81] Jussi Tamminen and Tuomas Koiranen. Mixing performance comparison of milliscale continuous high-shear mixers. *The Canadian Journal of Chemical Engineering*, 93(12):2245–2252, dec 2015.

[82] Aditya U. Vanarase, Juan G. Osorio, and Fernando J. Muzzio. Effects of powder flow properties and shear environment on the performance of continuous mixing of pharmaceutical powders. *Powder Technology*, 246:63–72, sep 2013.

[83] Liwei Cao, Danilo Russo, Kobi Felton, Daniel Salley, Abhishek Sharma, Graham Keenan, Werner Mauer, Huanhuan Gao, Leroy Cronin, and Alexei A Lapkin. Optimization of formulations using robotic experiments driven by machine learning doe. *Cell Reports Physical Science*, 2(1):100295, 2021.

[84] A. J. Vegas and D. G. Anderson. High-Throughput Approaches. In *Polymer Science: A Comprehensive Reference, 10 Volume Set*, volume 9, pages 457–484. Elsevier, jan 2012.

[85] Shihua Li, Jing Yan, and Lingxi Li. Automated Guided Vehicle: The Direction of Intelligent Logistics. In *Proceedings of the 2018 IEEE International Conference on Service Operations and Logistics, and Informatics, SOLI 2018*, pages 250–255. Institute of Electrical and Electronics Engineers Inc., sep 2018.

[86] Angelika Weber, Erich Von Roedern, and Hans Ulrich Stilz. SynCar: An approach to automated synthesis. *Journal of Combinatorial Chemistry*, 7(2):178–184, mar 2005.

[87] Brice Martin Couillaud, Philippe Espeau, Nathalie Mignet, and Yohann Corvis. State of the Art of Pharmaceutical Solid Forms: from Crystal Property Issues to Nanocrystals Formulation. *ChemMedChem*, 14(1):8–23, jan 2019.

[88] Phuong Tran, Yong-Chul Pyo, Dong-Hyun Kim, Sang-Eun Lee, Jin-Ki Kim, and Jeong-Sook Park. Overview of the Manufacturing Methods of Solid Dispersion Technology for Improving the Solubility of Poorly Water-Soluble Drugs and Application to Anticancer Drugs. *Pharmaceutics*, 11(3):132, mar 2019.

[89] John Reader. Automation in Medicinal Chemistry. *Current Topics in Medicinal Chemistry*, 4(7):671–686, mar 2005.

[90] Adam M. Fermier, John Troisi, Erin C. Heritage, Melissa A. Drexel, Pablo Gallea, and Kelly A. Swinney. Powder dispensing robot for sample preparation. *Analyst*, 128(6):790–795, jun 2003.

[91] Stephen K-F Wong, YiFeng Lu, William Heineman, Janice Palmer, and Carter Courtney. Fully automated solid weighing workstation. *Journal of biomolecular screening*, 10(5):524–31, aug 2005.

[92] Lorenz M. Baumgartner, Connor W. Coley, Brandon J. Reizman, Kevin W. Gao, and Klavs F. Jensen. Optimum catalyst selection over continuous and discrete process variables with a single droplet microfluidic reaction platform. *Reaction Chemistry and Engineering*, 3(3):301–311, jun 2018.

[93] Lorenz M. Baumgartner, Joseph M. Dennis, Nicholas A. White, Stephen L. Buchwald, and Klavs F. Jensen. Use of a Droplet Platform to Optimize Pd-Catalyzed C-N Coupling Reactions Promoted by Organic Bases. *Organic Process Research and Development*, 23(8):1594–1601, aug 2019.

[94] Connor W. Coley, Milad Abolhasani, Hongkun Lin, and Klavs F. Jensen. Material-Efficient Microfluidic Platform for Exploratory Studies of Visible-Light Photoredox Catalysis. *Angewandte Chemie International Edition*, 56(33):9847–9850, aug 2017.

[95] Nicholas Holmes, Geoffrey R. Akien, A. John Blacker, Robert L. Woodward, Rebecca E. Meadows, and Richard A. Bourne. Self-optimisation of the final stage in the synthesis of EGFR kinase inhibitor AZD9291 using an automated flow reactor. *Reaction Chemistry and Engineering*, 1(4):366–371, aug 2016.

[96] Christopher A. Hone, Nicholas Holmes, Geoffrey R. Akien, Richard A. Bourne, and Frans L. Muller. Rapid multistep kinetic model generation from transient flow data. *Reaction Chemistry and Engineering*, 2(2):103–108, apr 2017.

[97] Christopher A. Hone, Alistair Boyd, Anne O'Kearney-Mcmullan, Richard A. Bourne, and Frans L. Muller. Definitive screening designs for multistep kinetic models in flow. *Reaction Chemistry and Engineering*, 4(9):1565–1570, sep 2019.

[98] Hsiao Wu Hsieh, Connor W. Coley, Lorenz M. Baumgartner, Klavs F. Jensen, and Richard I. Robinson. Photoredox Iridium-Nickel Dual-Catalyzed Decarboxylative Arylation Cross-Coupling: From Batch to Continuous Flow via Self-Optimizing Segmented Flow Reactor. *Organic Process Research and Development*, 22(4):542–550, apr 2018.

[99] K. Poscharny, D. C. Fabry, S. Heddrich, E. Sugiono, M. A. Liauw, and M. Rueping. Machine assisted reaction optimization: A self-optimizing reactor system for continuous-flow photochemical reactions. *Tetrahedron*, 74(25):3171–3175, jun 2018.

[100] Brandon J. Reizman and Klavs F. Jensen. Simultaneous solvent screening and reaction optimization in microliter slugs. *Chemical Communications*, 51(68):13290–13293, jul 2015.

[101] Brandon J. Reizman and Klavs F. Jensen. Feedback in Flow for Accelerated Reaction Development. *Accounts of Chemical Research*, 49(9):1786–1796, sep 2016.

[102] Brandon J. Reizman, Yi Ming Wang, Stephen L. Buchwald, and Klavs F. Jensen. Suzuki-Miyaura cross-coupling optimization enabled by automated feedback. *Reaction Chemistry and Engineering*, 1(6):658–666, dec 2016.

[103] Marc Rodriguez-Garcia, Andrew J. Surman, Geoffrey J.T. Cooper, Irene Suárez-Marina, Zied Hosni, Michael P. Lee, and Leroy Cronin. Formation of oligopeptides in high yield under simple programmable conditions. *Nature Communications*, 6(1):1–7, oct 2015.

[104] Conor Waldron, Arun Pankajakshan, Marco Quaglio, Enhong Cao, Federico Galvanin, and Asterios Gavriilidis. An autonomous microreactor platform for the rapid identification of kinetic models. *Reaction Chemistry and Engineering*, 4(9):1623–1636, sep 2019.

[105] Cheng Zhu, Keshav Raghuvanshi, Connor W. Coley, Dawn Mason, Jody Rodgers, Mesfin E. Janka, and Milad Abolhasani. Flow chemistry-enabled studies of rhodium-catalyzed hydroformylation reactions. *Chemical Communications*, 54(62):8567–8570, jul 2018.

[106] Alexander Echtermeyer, Yehia Amar, Jacek Zakrzewski, and Alexei Lapkin. Self-optimisation and model-based design of experiments for developing a C-H activation flow process. *Beilstein J. Org. Chem*, 13:150–163, 2017.

[107] Jarosław M Granda, Liva Donina, Vincenza Dragone, De-Liang Long, and Leroy Cronin. Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature*, 559(7714):377–381, jul 2018.

[108] Jason S. Moore, Christopher D. Smith, and Klavs F. Jensen. Kinetics analysis and automated online screening of aminocarbonylation of aryl halides in flow. *Reaction Chemistry and Engineering*, 1(3):272–279, jun 2016.

[109] Robert W. Epps, Kobi C. Felton, Connor W. Coley, and Milad Abolhasani. Automated microfluidic platform for systematic studies of colloidal perovskite nanocrystals: Towards continuous nano-manufacturing. *Lab on a Chip*, 17(23):4040–4047, nov 2017.

[110] Pascal Neumann, Liwei Cao, Danilo Russo, Vassilios S. Vassiliadis, and Alexei A. Lapkin. A new formulation for symbolic regression to identify physico-chemical laws from experimental data. *Chemical Engineering Journal*, page 123412, nov 2019.

[111] Liwei Cao, Danilo Russo, Werner Mauer, Huan Huan Gao, and Alexei A Lapkin. Machine learning-aided process design for formulated products. In *Computer Aided Chemical Engineering*, volume 48, pages 1789–1794. Elsevier, 2020.

[112] Suraj Deshmukh, Matthew T. Bishop, Daniel Dermody, Laura Dietsche, Tzu Chi Kuo, Melissa Mushrush, Keith Harris, Jonathan Zieman, Paul Morabito, Brian Orvosh, and Don Patrick. A Novel High-Throughput Viscometer. *ACS Combinatorial Science*, 18(7):405–414, jul 2016.

[113] Jörg Läuger and Michael Krenn. From sample changer to the robotic rheometer: Automation and high throughput screening in rotational rheometry. In *AIP Conference Proceedings*, volume 1027, pages 1198–1201. American Institute of Physics, 2008.

[114] Bronwyn Ormsby, Alan Phenix, Melinda Keefe, and Tom Learner. A productive collaboration between conservation and industry: developing wet surface cleaning systems for unvarnished painted surfaces. *Studies in Conservation*, 61(sup2):313–314, 2016.

[115] Sushant S. Garud, Iftekhar A. Karimi, and Markus Kraft. Design of computer experiments: A review, nov 2017.

[116] A. Baroutaji, E. Morris, and A. G. Olabi. Quasi-static response and multi-objective crashworthiness optimization of oblong tube under lateral loading. *Thin-Walled Structures*, 82:262–277, sep 2014.

[117] Razi Sheikholeslami and Saman Razavi. Progressive Latin Hypercube Sampling: An efficient approach for robust sampling-based analysis of environmental models. *Environmental Modelling and Software*, 93:109–126, jul 2017.

[118] Jack P. C. Kleijnen. Design and Analysis of Monte Carlo Experiments. In *Handbook of Computational Statistics*, pages 529–547. Springer Berlin Heidelberg, 2012.

[119] James E. Gentle, Wolfgang Karl Härdle, and Yuichi Mori. *Handbook of computational statistics: Concepts and methods: Second Edition*. Springer Berlin Heidelberg, jan 2012.

[120] Shaul Mordechai. *Applications of Monte Carlo Method in Science and Engineering*. InTech, mar 2012.

[121] C Robert and G Casella. *Monte Carlo Statistical Methods*. Springer Science & Business Media, New York, 2013.

[122] K.T. Fang, R. Li, and A. Sudyianto. *Design and Modeling for Computer Experiments*. CRC press, New York, 2005.

[123] Fred J Hickernell. Lattice rules: how well do they measure up? In *Random and quasi-random point sets*, pages 109–166. Springer, 1998.

[124] V. Eglajs and P. Audze. New approach to the design of multifactor experiments. *Problems of Dynamics and Strengths*, 35(1):104–107, 1977.

[125] M. E. Johnson, L. M. Moore, and D. Ylvisaker. Minimax and maximin distance designs. *Journal of Statistical Planning and Inference*, 26(2):131–148, oct 1990.

[126] Des Raj. *Sampling theory*. McGraw-Hill, New York, 1968.

[127] Boxin Tang. Orthogonal array-based latin hypercubes. *Journal of the American Statistical Association*, 88(424):1392–1397, 1993.

[128] Kenny Q. Ye. Orthogonal Column Latin Hypercubes and Their Application in Computer Experiments. *Journal of the American Statistical Association*, 93(444):1430–1439, dec 1998.

[129] D. V. Lindley. On a measure of the information provided by an experiment. *The Annals of Mathematical Statistics*, pages 986–1005, 1956.

[130] C. E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, jul 1948.

[131] SACKS and J. Spatial designs. *Statistical design theory and related topics IV-2*, pages 385–399, 1988.

[132] M C Shewry and H P Wynn. Maximum entropy sampling. *Journal of Applied Statistics*, 14(2):165–170, 1987.

[133] Jerome Sacks, Susannah B. Schiller, and William J. Welch. Designs for computer experiments. *Technometrics*, 31(1):41–47, 1989.

[134] G. E. Box and N. R. Draper. A basis for the selection of a response surface design. *Journal of the American Statistical Association*, 54(287):622–654, 1959.

[135] Karel Crombecq, Dirk Gorissen, Dirk Deschrijver, and Tom Dhaene. A novel hybrid sequential design strategy for global surrogate modeling of computer experiments. *SIAM Journal on Scientific Computing*, 33(4):1948–1974, aug 2011.

[136] Qi Zhou, Xinyu Shao, Ping Jiang, Hui Zhou, and Leshi Shu. An adaptive global variable fidelity metamodeling strategy using a support vector regression based scaling function. *Simulation Modelling Practice and Theory*, 59:18–35, dec 2015.

[137] Qi Zhou, Xinyu Shao, Ping Jiang, Zhongmei Gao, Hui Zhou, and Leshi Shu. An active learning variable-fidelity metamodelling approach based on ensemble of metamodels and objective-oriented sequential sampling. *Journal of Engineering Design*, 27(4-6):205–231, may 2016.

[138] John Eason and Selen Cremaschi. Adaptive sequential sampling for surrogate model generation with artificial neural networks. *Computers and Chemical Engineering*, 68:220–232, sep 2014.

[139] Fani Boukouvala and Marianthi G. Ierapetritou. Surrogate-based optimization of expensive flowsheet modeling for continuous pharmaceutical manufacturing. *Journal of Pharmaceutical Innovation*, 8(2):131–145, jun 2013.

[140] Rommel G. Regis. Constrained optimization by radial basis function interpolation for high-dimensional expensive black-box problems with infeasible initial points. *Engineering Optimization*, 46(2):218–243, feb 2014.

[141] Ky Khac Vu, Claudia D'Ambrosio, Youssef Hamadi, and Leo Liberti. Surrogate-based methods for black-box optimization. *International Transactions in Operational Research*, 24(3):393–424, may 2017.

[142] Alison Cozad, Nikolaos V Sahinidis, and David C Miller. A combined first-principles and data-driven approach to model building. *Computers & Chemical Engineering*, 73:116–127, 2015.

[143] Zachary T. Wilson and Nikolaos V. Sahinidis. The ALAMO approach to machine learning. *Computers & Chemical Engineering*, 106:785–795, nov 2017.

[144] Claudia Houben, Nicolai Peremezhney, Alexandr Zubov, Juraj Kosek, and Alexei A. Lapkin. Closed-Loop Multitarget Optimization for Discovery of New Emulsion Polymerization Recipes. *Organic Process Research and Development*, 19(8):1049–1053, mar 2015.

[145] Ajith Abraham and Lakhmi Jain. Evolutionary Multiobjective Optimization. In *Evolutionary Multiobjective Optimization*, pages 1–6. Springer-Verlag, sep 2005.

[146] Eric Bradford, Artur M. Schweidtmann, and Alexei Lapkin. Efficient multiobjective optimization employing Gaussian processes, spectral sampling and a genetic algorithm. *Journal of Global Optimization*, 71(2):407–438, jun 2018.

[147] Johanna Kleinekorte, Lorenz Fleitmann, Marvin Bachmann, Arne Kätelhön, Ana Barbosa-Póvoa, Niklas von der Assen, and André Bardow. Life Cycle Assessment for the Design of Chemical Processes, Products, and Supply Chains. *Annual review of chemical and biomolecular engineering*, 11:203–233, jun 2020.

[148] Yehia Amar, Artur M Schweidtmann, Paul Deutsch, Liwei Cao, and Alexei Lapkin. Machine learning and molecular descriptors enable rational solvent selection in asymmetric catalysis. *Chemical science*, 10(27):6697–6706, 2019.

[149] Seyed Mohamad Moosavi, Arunraj Chidambaram, Leopold Talirz, Maciej Haranczyk, Kyriakos C. Stylianou, and Berend Smit. Capturing chemical intuition in synthesis of metal-organic frameworks. *Nature Communications*, 10(1):1–7, dec 2019.

[150] Daniel Salley, Graham Keenan, Jonathan Grizou, Abhishek Sharma, Sergio Martín, and Leroy Cronin. A nanomaterials discovery robot for the darwinian evolution of shape programmable gold nanoparticles. *Nature Communications*, 11(1):1–7, 2020.

[151] Liwei Cao, Danilo Russo, and Alexei A Lapkin. Automated robotic platforms in design and development of formulations. *AIChE Journal*, 67(5):e17248, 2021.

[152] Daniel S Salley, Graham A Keenan, De-Liang Long, Nicola L Bell, and Leroy Cronin. A modular programmable inorganic cluster discovery robot for the discovery and synthesis of polyoxometalates. *ACS central science*, 6(9):1587–1593, 2020.

[153] Ben GB Kitchener, John Wainwright, and Anthony J Parsons. A review of the principles of turbidity measurement. *Progress in Physical Geography*, 41(5):620–642, 2017.

[154] David D. Lewis and William A. Gale. A sequential algorithm for training text classifiers. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 1994*, pages 3–12. Association for Computing Machinery, Inc, aug 1994.

[155] Pablo Ruiz, Javier Mateos, Gustavo Camps-Valls, Rafael Molina, and Aggelos K Katsaggelos. Bayesian active remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 52(4):2186–2196, 2013.

[156] Freddy Kurniawan et al. Automatic music classification for dangdut and campursari using naïve bayes. In *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics*, pages 1–6. IEEE, 2011.

[157] Bo Tang, Haibo He, Paul M Baggenstoss, and Steven Kay. A bayesian classification approach using class-specific features for text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 28(6):1602–1606, 2016.

[158] Harry Zhang. The optimality of naive bayes. In *Artificial Intelligence Research Society Conference*, Florida, 2004.

[159] Jonathan Huang. Maximum likelihood estimation of dirichlet distribution parameters. *CMU Technique Report*, 2005.

[160] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

[161] Michael Emmerich, Kaifeng Yang, André Deutz, Hao Wang, and Carlos M Fonseca. A multicriteria generalization of bayesian global optimization. In *Advances in Stochastic and Deterministic Global Optimization*, pages 229–242. Springer, 2016.

[162] Pritam Ranjan and Neil Spencer. Space-filling Latin hypercube designs based on randomization restrictions in factorial experiments. *Statistics and Probability Letters*, 94:239–247, nov 2014.

[163] Tadayoshi Fushiki. Estimation of prediction error by using K-fold cross-validation. *Stat Comput*, 21:137–146, 2011.

[164] Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *Science*, 324(5923):81–85, apr 2009.

[165] Silviu Marian Udrescu and Max Tegmark. AI Feynman: A physics-inspired method for symbolic regression. *Science Advances*, 6(16):eaay2631, apr 2020.

[166] Michele Mattei, Georgios M Kontogeorgis, and Rafiqul Gani. A comprehensive framework for surfactant selection and design for emulsion based chemical product design. *Fluid Phase Equilibria*, 362:288–299, 2014.

[167] Genyuan Li, Caleb Bastian, William Welsh, and Herschel Rabitz. Experimental design of formulations utilizing high dimensional model representation. *The Journal of Physical Chemistry A*, 119(29):8237–8249, 2015.

[168] Abolghasem Jouyban, Hak-Kim Chan, Nora Yat Knork Chew, Maryam Khoubnasabjafari, and William Eugene Acree Jr. Solubility prediction of paracetamol in binary and ternary solvent mixtures using jouyban–acree model. *Chemical and pharmaceutical bulletin*, 54(4):428–431, 2006.

[169] Abolghasem Jouyban. Review of the cosolvency models for predicting solubility of drugs in water-cosolvent mixtures. *Journal of Pharmacy & Pharmaceutical Sciences*, 11(1):32–58, 2008.

[170] Abolghasem Jouyban, Ali Shayanfar, Vahid Panahi-Azar, Jafar Soleymani, BehroozH Yousefi, WilliamE Acree Jr, and Peter York. Solubility prediction of drugs in mixed solvents using partial solubility parameters. *Journal of pharmaceutical sciences*, 100(10):4368–4382, 2011.

[171] Mariano Martín and Alberto Martínez. A methodology for simultaneous product and process design in the customer products industry: The case study of the laundry business. In *Computer Aided Chemical Engineering*, volume 32, pages 715–720. Elsevier, 2013.

[172] FP Bernardo. Integrated process and product design optimization. In *Computer Aided Chemical Engineering*, volume 39, pages 347–372. Elsevier, 2016.

[173] Ignacio E Grossmann. Challenges in the new millennium: product discovery and design, enterprise and supply chain optimization, global life cycle assessment. *Computers & Chemical Engineering*, 29(1):29–39, 2004.

[174] Bradley Jones, Rachel T. Silvestrini, Douglas C. Montgomery, and David M. Steinberg. Bridge designs for modeling systems with low noise. *Technometrics*, 57(2):155–163, apr 2015.

[175] Elizabeth G. Ryan, Christopher C. Drovandi, James M. McGree, and Anthony N. Pettitt. A Review of Modern Computational Algorithms for Bayesian Optimal Design. *International Statistical Review*, 84(1):128–154, apr 2016.

[176] Kathryn Chaloner and Isabella Verdinelli. Bayesian experimental design: A review. *Statistical Science*, pages 273–304, 1995.

[177] Antony M. Overstall and David C. Woods. Bayesian Design of Experiments Using Approximate Coordinate Exchange. *Technometrics*, 59(4):458–470, oct 2017.

[178] Valerii Vadimovich Fedorov. *Theory of optimal experiments*. Elsevier, 2013.

[179] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient Global Optimization of Expensive Black-Box Functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

[180] Nikolai V Smirnov. Estimate of deviation between empirical distribution functions in two independent samples. *Bulletin Moscow University*, 2(2):3–16, 1939.

[181] Jonas Mockus. *The Application Of Bayesian Methods*. Elsevier, North Holland, Amsterdam, 1989.

[182] Yoav Benjamini. Simultaneous and selective inference: Current successes and future challenges. *Biometrical Journal*, 52(6):708–721, 2010.

[183] Nor Alafiza Yunus, Krist V Gernaey, John M Woodley, and Rafiqul Gani. A systematic methodology for design of tailor-made blended products. *Computers & chemical engineering*, 66:201–213, 2014.

[184] C Couteau, H Diarra, and L Coiffard. Effect of the product type, of the amount of applied sunscreen product and the level of protection in the uvb range on the level of protection achieved in the uva range. *International Journal of Pharmaceutics*, 500(1-2):210–216, 2016.

[185] Nicolai Peremezhney, Colm Connaughton, Gianfranco Unali, Evor Hines, and Alexei A Lapkin. Application of dimensionality reduction to visualisation of high-throughput data and building of a classification model in formulated consumer product design. *Chemical Engineering Research and Design*, 90(12):2179–2185, 2012.

[186] K Hoffmann, K Kaspar, P Altmeyer, and T Gambichler. Uv transmission measurements of small skin specimens with special quartz cuvettes. *Dermatology*, 201(4):307–311, 2000.

[187] John D'Orazio, Stuart Jarrett, Alexandra Amaro-Ortiz, and Timothy Scott. Uv radiation and the skin. *International journal of molecular sciences*, 14(6):12222–12248, 2013.

[188] Thomas Schwarz. Mechanisms of uv-induced immunosuppression. *The Keio journal of medicine*, 54(4):165–171, 2005.

[189] Steven Q Wang, Richard Setlow, Marianne Berwick, David Polsky, Ashfaq A Marghoob, Alfred W Kopf, and Robert S Bart. Ultraviolet a and melanoma: a review. *Journal of the American Academy of Dermatology*, 44(5):837–846, 2001.

[190] Michael F Holick. Sunlight, uv-radiation, vitamin d and skin cancer: how much sunlight do we need? In *Sunlight, vitamin D and skin cancer*, pages 1–15. Springer, 2008.

[191] Bernd Herzog and Uli Osterwalder. Simulation of sunscreen performance. *Pure and Applied Chemistry*, 87(9-10):937–951, 2015.

[192] L Mbanga, M Mulenga, PT Mpiana, K Bokolo, M Mumbwa, and K Mvingu. Determination of sun protection factor (spf) of some body creams and lotions marketed in kinshasa by ultraviolet spectrophotometry. *International Journal of Advanced Research in Chemical Science (IJARCS)*, 1(8):7–13, 2014.

[193] Jiyong Shim, Jun Man Lim, and Sun Gyoo Park. Machine learning for the prediction of sunscreen sun protection factor and protection grade of uva. *Experimental dermatology*, 28(7):872–874, 2019.

[194] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* Adaptive Computation and Machine Learning series, 2018.

[195] Daniel T Larose and Chantal D Larose. *Discovering knowledge in data: an introduction to data mining*, volume 4. John Wiley & Sons, 2014.

[196] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[197] Chong Ho Yu. Abduction? deduction? induction? is there a logic of exploratory data analysis?. In *Annual Meeting of American Educational Research Association,New Orleans, Louisiana*. ERIC, 1994.

[198] Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. Pearson correlation coefficient. In *Noise reduction in speech processing*, pages 1–4. Springer, 2009.

[199] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[200] Sebastian Raschka. *Python machine learning*. Packt publishing ltd, 2015.

[201] Ian T Jolliffe and Jorge Cadima. Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202, 2016.

[202] Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[203] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[204] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[205] Kilian Q Weinberger and Gerald Tesauro. Metric learning for kernel regression. In *Artificial Intelligence and Statistics*, pages 612–619, 2007.

[206] Leo Breiman, Jerome Friedman, Charles J Stone, and Richard A Olshen. *Classification and regression trees*. CRC press, 1984.

[207] RK Brouwer. A hybrid neural network with fuzzy rules for categorical and numeric input. In *IEEE Annual Meeting of the Fuzzy Information, 2004. Processing NAFIPS'04.*, volume 1, pages 319–324. IEEE, 2004.

[208] Nicoleta Rogovschi, Mustapha Lebbah, and Younès Bennani. Probabilistic mixed topological map for categorical and continuous data. In *2008 Seventh International Conference on Machine Learning and Applications*, pages 224–231. IEEE, 2008.

[209] Cedric Seger. An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing, 2018.

[210] G. Skoraczyński, P. Dittwald, B. Miasojedow, S. Szymkuć, E. P. Gajewska, B. A. Grzybowski, and A. Gambin. Predicting the outcomes of organic reactions via machine learning: are current descriptors sufficient? *Scientific Reports*, 7(1):3582, dec 2017.

[211] Brandon J. Reizman and Klavs F. Jensen. Feedback in Flow for Accelerated Reaction Development. *Accounts of Chemical Research*, 49(9):1786–1796, sep 2016.

[212] Olaf Wolkenhauer. Why model? *Frontiers in Physiology*, 5:21, jan 2014.

[213] Alexei A Lapkin, Adelina Voutchkova, and Paul Anastas. A conceptual framework for description of complexity in intensive chemical processes. *Chemical Engineering and Processing: Process Intensification*, 50(10):1027–1034, 2011.

[214] Alison Cozad, Nikolaos V. Sahinidis, and David C. Miller. Learning surrogate models for simulation-based optimization. *AIChE Journal*, 60(6):2211–2227, jun 2014.

[215] Shanwu Li, Eurika Kaiser, Shujin Laima, Hui Li, Steven L Brunton, and J Nathan Kutz. Discovering time-varying aeroelastic models of a long-span suspension bridge from field measurements by sparse identification of nonlinear dynamical systems. *arXiv preprint arXiv:1809.05707*, 2018.

[216] Niall M Mangan, J Nathan Kutz, Steven L Brunton, and Joshua L Proctor. Model selection for dynamical systems via sparse regression and information criteria. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2204):20170009, 2017.

[217] Abhinav Narasingam and Joseph Sang-Il Kwon. Data-driven identification of interpretable reduced-order models using sparse regression. *Computers & Chemical Engineering*, 119:101–111, nov 2018.

[218] Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4):e1602614, apr 2017.

[219] Hayden Schaeffer. Learning partial differential equations via data discovery and sparse optimization. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2197):20160446, 2017.

[220] Sheng Zhang and Guang Lin. Robust data-driven discovery of governing physical laws with error bars. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 474(2217):20180305, sep 2018.

[221] Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*, 113(15):3932–7, apr 2016.

[222] Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.

[223] Bashar Tarawneh, Wassel AL Bodour, and Khaled Al Ajmi. Intelligent Computing Based Formulas to Predict the Settlement of Shallow Foundations on Cohesionless Soils. *The Open Civil Engineering Journal*, 13(1):1–9, feb 2019.

[224] Yiqun Wang, Nicholas Wagner, and James M Rondinelli. Symbolic regression in materials science. *MRS Communications*, 9(3):793–805, 2019.

[225] Vassilios S. Vassiliadis, Yian Wang, Harvey Arellano-Garcia, and Ye Yuan. A Novel Rigorous Mathematical Programming Approach to Construct Phenomenological Models. *Computer Aided Chemical Engineering*, 37:707–712, jan 2015.

[226] Alison Cozad and Nikolaos V Sahinidis. A global minlp approach to symbolic regression. *Mathematical Programming*, 170(1):97–119, 2018.

[227] Ignacio E Grossmann. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and engineering*, 3(3):227–252, 2002.

[228] Jan Kronqvist, David E Bernal, Andreas Lundell, and Ignacio E Grossmann. A review and comparison of solvers for convex minlp. *Optimization and Engineering*, 20(2):397–455, 2019.

[229] Ruth Misener and Christodoulos A. Floudas. ANTIGONE: Algorithms for coNTinuous / Integer Global Optimization of Nonlinear Equations. *Journal of Global Optimization*, 59(2-3):503–526, jul 2014.

[230] Mustafa R. Kılınç and Nikolaos V. Sahinidis. Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems with BARON. *Optimization Methods and Software*, 33(3):540–562, may 2018.

[231] Pietro Belotti, Jon Lee, Leo Liberti, François Margot, and Andreas Wächter. Branching and bounds tighteningtechniques for non-convex MINLP. *Optimization Methods and Software*, 24(4-5):597–634, oct 2009.

[232] Youdong Lin and Linus Schrage. The global solver in the LINDO API. *Optimization Methods and Software*, 24(4-5):657–668, oct 2009.

[233] Stefan Vigerske and Ambros Gleixner. SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework. *Optimization Methods and Software*, 33(3):563–593, may 2018.

[234] CPLEX User's Manual. Ibm ilog cplex optimization studio. *Version*, 12:1987–2020, 1987.

[235] William E. Hart, Jean-Paul Watson, and David L. Woodruff. Pyomo: modeling and solving mathematical programs in Python. *Mathematical Programming Computation*, 3(3):219–260, sep 2011.

[236] Aaron Meurer, Christopher P. Smith, Mateusz Paprocki, Ondřej Čertík, Sergey B. Kirpichev, Matthew Rocklin, AMiT Kumar, Sergiu Ivanov, Jason K. Moore, Sartaj Singh, Thilina Rathnayake, Sean Vig, Brian E. Granger, Richard P. Muller, Francesco Bonazzi, Harsh Gupta, Shivam Vats, Fredrik Johansson, Fabian Pedregosa, Matthew J. Curry, Andy R. Terrel, Štěpán Roučka, Ashutosh Saboo, Isuru Fernando, Sumith Kulal, Robert Cimrman, and Anthony Scopatz. SymPy: symbolic computing in Python. *PeerJ Computer Science*, 3:e103, jan 2017.

[237] Markus Quade, Markus Abel, Kamran Shafi, Robert K Niven, and Bernd R Noack. Prediction of dynamical systems by symbolic regression. *PHYSICAL REVIEW E*, 94:12214, 2016.

[238] Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 1974.

[239] Scott I. Vrieze. Model selection and psychological theory: A discussion of the differences between the Akaike information criterion (AIC) and the Bayesian information criterion (BIC). *Psychological Methods*, 17(2):228–243, 2012.

[240] Christopher W Macosko and Ronald G Larson. *Rheology: principles, measurements, and applications*. Vch New York, 1994.

[241] Faith A Morrison et al. *Understanding rheology*. Oxford University Press, USA, 2001.

[242] Raj P Chhabra and John Francis Richardson. *Non-Newtonian flow and applied rheology: engineering applications*. Butterworth-Heinemann, 2011.

[243] Andreas Volk and Christian J. Kähler. Density model for aqueous glycerol solutions. *Experiments in Fluids*, 59(5):75, may 2018.

[244] Nian Sheng Cheng. Formula for the viscosity of a glycerol-water mixture. *Industrial and Engineering Chemistry Research*, 47(9):3285–3288, 2008.

[245] Jorg Klausen, Markus A Meier, and Rene P Schwarzenbach. Assessing the fate of organic contaminants in aquatic environments: Mechanism and kinetics of hydrolysis of a carboxylic ester. *Journal of chemical education*, 74(12):1440, 1997.

[246] Peter S. Marrs. Class Projects in Physical Organic Chemistry: The Hydrolysis of Aspirin. *Journal of Chemical Education*, 81(6):870–873, 2004.