

New Technology for the Controlled Synthesis of Functional Materials in Flow

Submitted for the degree of
Doctor of Philosophy

Barnaby Walker
Centre for Plastic Electronics,
Department of Chemistry, Imperial College London
CID: 00548532

Supervisor:
Prof. John C. de Mello

6th December 2017

Copyright Licence

The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work

Statement of Originality

Except where specific reference is made, the material contained in this thesis is the result of my own work and has not already been submitted, either wholly or in part, to satisfy any degree requirement at this or any other university. Where work was the result of collaboration, or not wholly my own, appropriate attribution has been clearly indicated.

Barnaby Walker

Abstract

Functional materials - materials designed to perform a specific function - have many promising, novel properties that are dependent on their structure. Traditional synthetic methods typically do not provide the control and uniformity required to produce these materials in large quantities while maintaining strict control over their structure and purity. Flow chemistry has previously been used to gain improved control over the synthesis of functional materials, but has mostly been applied to single-step syntheses of a limited range of materials. The work carried out in this thesis focused on advancing flow chemistry technology for the synthesis of high quality functional materials in two key areas: self-optimising reactors, and synthesis in a three-phase flow reactor.

In the case of self-optimising reactors, an automated flow reactor was constructed for the synthesis of *o*-xylenyl functionalised fullerene derivatives. The cascadic nature of the synthesis results in a product consisting of a mixture of desirable and undesirable adducts, therefore making it a good test case for attempting to maximise the formation of one product while suppressing the formation of others. The automated flow reactor developed for this synthesis was composed of self-built components and incorporated in-line product analysis by high performance liquid chromatography. It was able to produce a stable, reproducible product over more than 50 hours, and proved capable of carrying out automated runs comprising over 240 separate measurements.

To control the automated flow reactor I developed an intuitive method for optimising competing requirements in a synthesis by formulating chemical optimisations as constrained optimisations. The requirements of a synthesis were encoded in a figure of merit designed to minimise a lead property while at the same time applying a penalty to violations of constraints on other, competing, properties. After testing different specifications, the synthesis of *o*-xylenyl functionalised fullerene derivatives was optimised successfully to yield a product with the best balance of the two desirable products, X_2 and X_1 , while suppressing the amount of the least desirable product, X_3 , to below 10 %.

Three-phase reactor technology was applied to the controlled synthesis of magnetic iron oxide nanoparticles, via thermal decomposition of $\text{Fe}(\text{acac})_3$. Although this synthetic route should give more control over nanoparticle size and size uniformity than other syntheses, previous work on adapting it to flow had demonstrated that there was a large problem with reactor fouling due to product deposition in the reaction channel. Three-phase flow was employed to solve this by spacing the reaction droplets out from each other to prevent agglomeration. After changing the reaction solvent to improve the flow characteristics of the reaction mixture, a working three-phase flow synthesis of iron oxide nanoparticles was demonstrated without any fouling.

Acknowledgements

Doing a PhD is long. I've been lucky enough to have people in my life that have made everything a bit easier, and so I would like to thank them here.

Perhaps most of all, I need to thank my supervisor, Prof. John de Mello for all of the help and support he has given me. Your advice and guidance has been invaluable and was (usually) always right. I'm sorry I didn't visit you more often.

I'd also like to thank past and present members of the John de Mello group for making life in the lab more bearable. Both Dr. James Bannock and Dr. Thomas Phillips were always there to help when I was stuck, and their constant criticism always drove me to do better. They're two of the most genuine people that I've met, and my PhD would have been a lot less enjoyable without them. I'm also very grateful to Dr. Adrian Nightingale, who very patiently introduced me to flow chemistry and research in general, and to Dr. Siva Krishnadasan, for all the help he gave me in the lab. I also can't forget to thank Wenmin Xu, the best social secretary the group has ever had.

My work really wouldn't have got very far without Simon Turner, who made parts and prototypes used in this thesis, and who stopped me from electrocuting myself with bad wiring. I'm incredibly grateful for that.

I should probably also thank my parents. They've always given me support and encouragement throughout my studies, and really my whole life. They're pretty super. They were also very good at reminding me that I need to finish my thesis once a week for the past year.

Above all I'd like to thank Moons for keeping me sane through all of it. I can't put into words how much you mean to me and just how much I owe you, but I hope that you know. I'm very much looking forward to all the things we can do now that neither of us has to write a thesis!

List of Publications

Tuning Reaction Products by Constrained Optimisation

B. E. Walker, J. H. Bannock, A. M. Nightingale, J. C. de Mello, *React. Chem. Eng.*, 2017, **2**, 785-798

Flow Synthesis of Electronic Materials

B. E. Walker, *Rank Prize Fund Symposium*, Grasmere, November 2015, Oral Presentation

Fullerenes in Flow: Continuous Processing for Cost-Effective Functional Materials

B. E. Walker, *12th International Conference on Materials Chemistry (MC12)*, York, 2015, Oral Presentation

List of Abbreviations

ANOVA analysis of variance
DAQ data acquisition
DMF dimethylformamide
EGFR epidermal growth factor inhibitor
ETFE ethylene tetrafluoroethylene
HPLC high performance liquid chromatography
HSV Hue Saturation Value
IR infra-red
LED light-emitting diode
ISPR longitudinal surface plasmon resonance
MSE mean squared error
NMR nuclear magnetic resonance
NP nanoparticle
***o*-DCB** *o*-dichlorobenzene
ODE 1-octadecene
OPV organic photovoltaics
***o*-QDM** *o*-quiodimethane
P3HT poly(3-hexylthiophene)
PDMS polydimethylsiloxane
PEEK polyether ether ketone
PFA perfluoroalkoxy
PID proportional-integral-derivative
PTFE polytetrafluoroethylene
RBG Red Blue Green
SEM scanning electron microscopy
SNOBfit Stable Noisy Optimisation by Branch and Fit
TEM transmission electron microscopy
UV ultra-violet
VSM vibrating sample magnetometry

Contents

Copyright Licence	3
Statement of Originality	5
Abstract	7
Acknowledgements	9
List of Publications	11
List of Abbreviations	13
1 Introduction and Background	19
1.1 An Introduction to Flow Chemistry	21
1.2 The Challenges of Flow Chemistry	23
1.2.1 Designing a Flow Reactor	23
1.2.2 Reactor Fouling	26
1.3 Automation and Optimisations	28
1.3.1 In-line Analysis	29
1.3.2 Self-Optimisation	31
1.4 Conclusions	37
2 Experimental Methods	39
2.1 General Materials	41
2.2 Batch Synthesis of <i>o</i> -Xylenyl Fullerene Adducts	41
2.2.1 Sultine Synthesis	41
2.2.2 Sultine Purity Estimation	41
2.2.3 Batch Synthesis of Fullerene Derivatives	43
2.2.4 Separation of Fullerene Derivatives	44
2.3 Construction of an Automated Flow Reactor	44
2.3.1 Development of a Solid-State Heater	45
2.3.2 HPLC System Development	50
2.3.3 Constructing the Complete Flow Reactor	62
2.4 Automated Flow Synthesis of Fullerene Adducts	63
2.4.1 Stock Solution Preparation	63
2.4.2 Automated Operation	64
2.4.3 Automatic Optimisation	64

2.5	Flow Synthesis of Iron Oxide Nanoparticles	65
2.5.1	Batch Synthesis of Iron Oxide Nanoparticles	65
2.5.2	Synthesis of Iron Oxide Nanoparticles in a Three-Phase Flow Reactor	66
2.5.3	Droplet Analysis	67
3	Automation and Optimisation of the Cascadic Synthesis of Fullerene Derivatives	69
3.1	Introduction	71
3.2	Flow Reactor Design and Automation	72
3.2.1	Choice of In-line Analysis	73
3.3	Automated Flow Synthesis	74
3.3.1	Reactor Stability and Reproducibility	75
3.3.2	Automated Reaction Condition Screening	80
3.4	Self-Optimisation	85
3.4.1	The Choice of Optimisation Algorithm	85
3.4.2	2D optimisation	86
3.4.3	3D optimisation	91
3.4.4	Changing Objective	95
3.5	Summary	97
4	Tuning Reaction Product Composition by Constrained Optimisation	99
4.1	Introduction	101
4.2	Constrained Optimisation	102
4.2.1	Changing sigma	108
4.2.2	Selecting Product Composition	117
4.3	Conclusions	129
5	Synthesis of Iron Oxide Nanoparticles in a Three-Phase Droplet Reactor	131
5.1	Introduction	133
5.1.1	Magnetic Properties of Iron Oxide Nanoparticles	133
5.1.2	Iron Oxide Nanoparticle Synthesis	135
5.1.3	Three-Phase Flow Synthesis of Iron Oxide Nanoparticles	138
5.1.4	Development of a Three-Phase Flow Synthesis of Iron Oxide Nanoparticles	139
5.2	Synthesis of Iron Oxide Nanoparticles in 2-Pyrrolidone	144
5.3	Synthesis of Iron Oxide Nanoparticles in Phenyl Ether	147
5.3.1	A Working Three-Phase Synthesis of Iron Oxide Nanoparticles . . .	149
5.4	Conclusion	158
5.4.1	Further Work	159
6	Conclusions and Further Work	161
6.1	Conclusions	163
6.2	Further Work	165
6.2.1	Automatic Optimisation of More Challenging Targets	165
6.2.2	Handling Uncertainty	166
6.2.3	Iron Oxide	167
	Appendix	169

CHAPTER 1

Introduction and Background

1.1 An Introduction to Flow Chemistry

The field of flow chemistry has received increasing attention over the past two decades, having been identified as an important enabling technology^{1,2}. A large amount of research in this area has focused on improving the yield or synthetic accessibility of pharmaceutically important small molecules, with notable successes being the continuous processing of molecules with multiple synthetic steps³, and the development of a platform for on-demand production of medicines⁴. However, the advantages of flow chemistry also make it a promising technology for improving the production of functional materials. Although what is defined as a functional material may somewhat depend the situation, in this thesis they are defined as materials designed to perform a specific physical function, such as absorbing or emitting light at a particular wavelength. This definition depends both on the product in question having an intrinsic property necessary to perform a particular function, and the product being used as a material. In doing so various nanoparticles and polymers are included in the definition, but a distinction is drawn between certain small molecules - for example, a dye molecule used as an electron acceptor in a thin film solar cell would be included, while a small bioactive molecule would not because it is used as a functional molecule rather than a material.

Functional materials are often challenging to make using traditional methods of synthesis. Because their properties - and therefore their suitability for specific applications - are dependent on their structure, strict control over synthetic conditions is necessary. For instance, the emission wavelength of fluorescent quantum dots depends on the diameter of the nanoparticles, which in turn is controlled by conditions during their synthesis, e.g. reaction time and temperature. However, traditional methods of synthesis often do not offer sufficient control over these parameters to reliably tune the properties of interest, especially when trying to produce large quantities of material.

The traditional method of synthesising chemicals, referred to here as “batch” chemistry, typically involves mixing reagents together in a reaction vessel such as a glass round-bottomed flask, and dissolving or suspending them in a common solvent. This reaction mixture is then stirred, with an external source of energy, like a heater, being used to drive the reaction if necessary. The use of reaction vessels like glass flasks can be seen as an advantage of batch chemistry, as the same vessel can be used for a range of different synthetic procedures. However, these vessels have large volumes compared to their surface areas, which can cause inefficient heat and mass transfer⁵. In consequence it is difficult to have a high degree of control over the reaction conditions, causing temperature and composition gradients throughout the reaction mixture. This in turn means that there will be inhomogeneous product formation and batch-to-batch variation in product quality. These problems only become worse upon scaling up to the large ($\gg 1$ L) reaction volumes needed for industrial manufacturing.

In contrast, flow chemistry utilises reaction vessels that are made of one or more channels with a typical width of 100 - 1000 μm ⁶ - straddling the range from micro to millifluidic reactors. This affords a large surface area to volume ratio, which creates much more efficient mixing and heating than in batch reactors. In a typical flow reaction, the reaction mixture is pumped through the reaction channel, where the reaction can be, for instance, heated. This means that every bit of the reaction mixture passes through the same reaction channel, and therefore, if the environmental conditions are held constant with time, experiences the same conditions. These benefits translate to a high degree of control over the reaction conditions and so allow for the production of high quality functional materials.

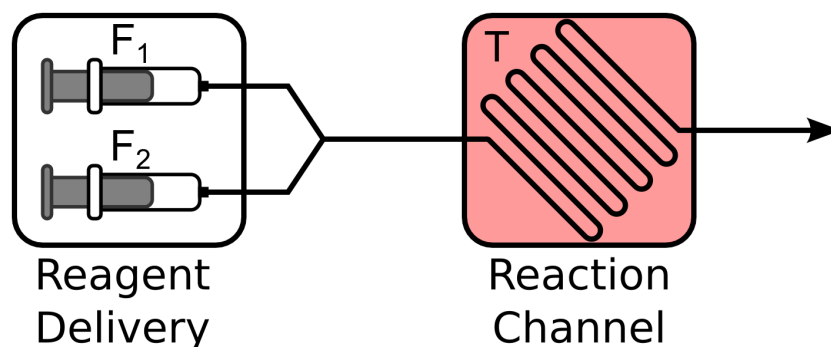


Figure 1.1: A simple schematic of a typical flow reactor showing reagent delivery from syringe pumps into a heated reaction channel.

The simple diagram in Fig. 1.1 shows an experimental set-up for a typical flow reaction, with syringe pumps delivering two reagents into a heated reaction channel. This gives control over three reaction parameters: the temperature of the reaction channel (T) can be controlled using a heater, the composition of the reaction mixture can be varied by changing the ratio of the two reagent flow rates (F_1 and F_2) to each other, and the average residence time can be varied by changing the overall flow rate (the sum of F_1 and F_2). As well as typically providing greater control, flow chemistry offers numerous advantages that make it an enabling technology for the synthesis of functional materials. Although high quality materials can be synthesised in batch on small scales, scaling up a batch process means increasing its volume, and therefore trading-off on the quality of the product due to increasingly poor heat and mass transfer. Flow syntheses do not suffer from this problem. There are two approaches to scaling up a flow synthesis: increasing the length of the reaction channel so higher flow rates may be used for the same overall residence time, or increasing the number of reaction channels that are fed from the same source of reagents. Neither of these approaches change the high surface area to volume ratio of the reaction channel, and therefore no trade off should be seen in the quality of the product when scaling up. Furthermore, given the continuous nature of a flow synthesis, if more product is needed, the reaction can just be run for longer.

Finally, the strict control over reaction conditions that flow chemistry provides, coupled with its continuous nature, means that reaction conditions can more easily be investigated and screened than in a batch process. For a batch synthesis, each set of reaction conditions being investigated needs to be run as a separate synthesis, with separate reagents and reaction vessels. For a flow synthesis, however, a single set of reagents can be used to feed a single flow reactor. Furthermore, by incorporating in-line analysis and interfacing the flow reactor with a computer, automated optimisations and high-throughput screening are made possible. This, in combination with the high degree of control over reaction conditions, means flow chemistry is potentially much better suited to the synthesis of high quality functional materials.

1.2 The Challenges of Flow Chemistry

Although flow chemistry has many advantages, as described above, there are a number of things to consider before employing this technology in the laboratory. An obvious consideration is whether or not to use flow chemistry in the first place. For some reactions, there is no clear benefit. For instance, if the synthesis proceeds quantitatively in batch (i.e. with good selectivity and few side products) it is doubtful that moving to flow will bring significant benefits. Similarly, some syntheses are not suitable for adapting to flow. For example, if the reactants form a slurry, pumping the reaction mixture may require excessive pressures or cause the flow reactor to block. Therefore consideration is needed both of what problem in a synthesis needs solving, and whether that problem can be addressed by adapting the synthesis to flow.

1.2.1 Designing a Flow Reactor

After deciding whether to use flow chemistry or not, the first consideration is how to design an appropriate flow reactor. A number of different reactor designs have been reported in the literature or are available from vendors, and it can be difficult to know which is the best to use. This is further complicated by the choice of whether to use a commercially available, off-the-shelf system, or to build a custom-designed flow reactor.

Buying a commercial flow system can save a lot of time, as they are usually simple to set up and use. Modular systems can even offer enough flexibility to carry out multi-step syntheses. However, the challenges of a particular synthesis, or even the details of a synthesis, are often unknown at the start of a project. Buying a complete system can lock the chemist into a specific reactor configuration that may ultimately not be suitable for the desired usage, or may work for one reaction but be unsuited to a future application. Given their significant cost, the lack of versatility is a real downside.

On the other hand, self-built systems offer almost limitless versatility and customis-

ability. Their major downfall is the time and know-how that it can take to get a reactor successfully up and running. For chemists that are new to flow chemistry, or have little experience of developing flow reactors, this can seem daunting. However, there has recently been a push from some researchers to provide detailed guidance on how and when to build your own flow reactors from readily available parts⁷. These go some way to making self-built flow reactors a more attractive option than a commercially bought solution, at least in the laboratory setting.

A number of flow syntheses have been reported by groups using commercially bought systems, although most of these appear to be for syntheses of pharmaceuticals and small molecules. Of particular note are the multi-step flow syntheses of a number of pharmaceutical and natural products by the Ley group^{8–15}, using a combination of commercial systems as well as flow equipment that they themselves have commercialised. This shows the relative ease with which fairly complicated multi-step chemistry can be accessed through the use of ready made flow solutions.

Interestingly there are only a small number of syntheses of functional materials found in the literature that use commercially bought flow systems. Two notable examples are the synthesis of the electroactive polymer poly(3-hexylthiophene)¹⁶ and the production of fullerene derivatives¹⁷, both reported by Seyler *et al.* Although these studies demonstrate the utility of commercial systems for the flow synthesis of some functional materials, it should be noted that the synthetic routes for the fullerene derivatives were similar, meaning little reactor customisation would have been necessary from one synthesis to the next. Furthermore, the synthesis of P3HT polymer makes no mention of reactor fouling, a problem that has been reported in other flow syntheses of polymers^{18–20} that would have required the customisation of the flow reactor to deal with.

Most of the literature examples of functional materials synthesised in flow used custom-built reactors. A number of different semiconductor^{6,21–26}, metal^{27–35}, and metal oxide^{22,36–41} nanoparticles have been synthesised in home-made flow reactors, as well as some electroactive polymers^{19,42,43}, metal-organic frameworks^{44,45}, and lead-halide perovskites⁴⁶. This range of different functional materials shows the versatility that comes with using a custom-built reactor. It may also be due to the high costs typical of commercial systems, which may be beyond the financial reach of many groups. Whatever the reasons, it seems clear that at the moment self-built flow reactors are preferred for the synthesis of functional materials.

Whether buying or building a flow reactor, a choice needs to be made about what type of reaction channel to use. There are two types of commonly employed channel⁶: a channel etched into a chip formed of a material like silicone or polydimethylsiloxane (PDMS), or a channel formed by small-diameter tubing made from silicone, fluorinated polymer, or steel. Chip-based reactors can offer exceptional control over reaction conditions, allowing the use of engineered channel patterns to manipulate the flow, and the incorporation of

multiple features onto one chip like multiple heating zones, temperature gradients, mixing areas, flow focusing, or excitation by a light source. However, the maximum volume, and therefore volumetric throughput, of chip-based reactors is limited by the typically small size of the chips. Furthermore, considerable effort and skill is required to pattern the chips, which may not make them accessible to first-time flow chemists.

Tube-based flow reactors are usually much easier to assemble. Construction of a tube-based reactor is often just a case of cutting tubes to length and joining them together. This also makes them very versatile, with changes to the channel architecture being simple to implement (as opposed to chip-based systems for which it is often necessary to design and etch a new chip). Similarly, reaction channel length is not limited by the size of a chip, so the maximum volumetric throughput of tube-based reactors is typically higher. For functional material synthesis, the versatility that tube-based flow reactors afford may be advantageous during development of a synthesis, as they can often present problems that are unknown in batch, e.g. unstable flow conditions as a result of increased viscosity or particle deposition as polymers and nanoparticles grow. The increased throughput is an important advantage, since it allows for the production of large amounts of high quality functional materials.

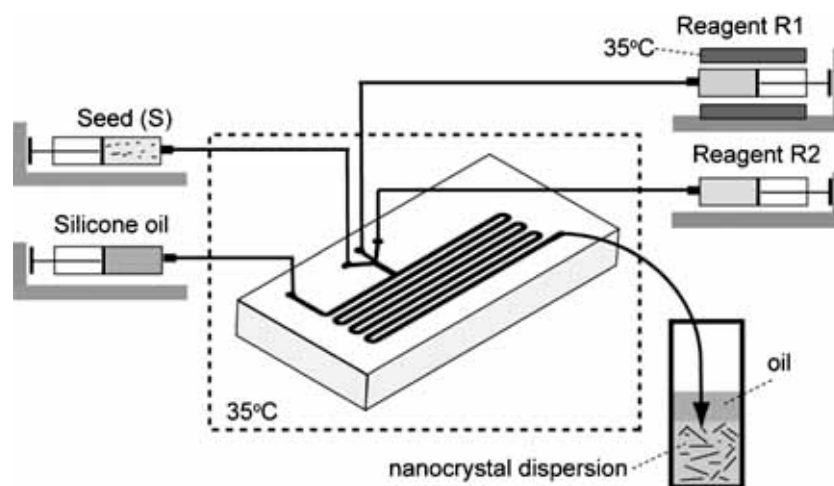


Figure 1.2: Schematic of a chip based reactor for the synthesis of anisotropic gold nanoparticles. Reproduced from Duraiswamy and Khan²⁸

Much of the work done on the flow synthesis of nanoparticles has been carried out in chip-based flow reactors. For example, Duraiswamy *et al* used the chip-based reactor illustrated in Fig. 1.2 to tune the size and aspect-ratio of gold nanorods²⁸. Similarly, Yang *et al* were able to synthesise blue-emitting core-shell quantum dots by using a microfluidic chip-reactor that incorporated an initial high temperature zone for nanoparticle nucleation, as well as a subsequent lower temperature zone for crystal growth⁴⁷. Clearly the high level of control over reaction conditions is a considerable advantage in these cases, with engineered channel patterns allowing very efficient reagent mixing and incorporation of different heating devices allowing separation of the nucleation and growth phases of

nanoparticle synthesis.

However, in recent years there has been an increase in the reported syntheses of functional materials using tube-based reactors. As well as the high temperature synthesis of a number of different nanoparticles²², syntheses of electroactive polymers^{16,18,19,42,43} and nanoscale polymer beads⁴⁸ have been reported. In fact, the flow synthesis of nanoscale polymer beads was reported in a scalable flow reactor (able to operate continuously over 24 hr) that could control the reaction conditions to tune the optical properties of the polymer nanoparticles⁴⁸. A scalable tube-based flow reactor has also been reported for the synthesis of CdTe quantum dots, achieving the largest reported rate of quantum dot synthesis (150 g/day) by using five identical reaction channels fed from the same reservoir of reaction mixture²⁴. These examples go some way to demonstrating that tube-based reactors may give more opportunity to scale up flow syntheses than chip-based reactors, with minimal trade-off in the control over reaction conditions seen for chip-based reactors. Given the relative simplicity and re-usability of tube-based reactors as well, it seems that they may be the preferable choice for most functional materials syntheses.

1.2.2 Reactor Fouling

Another challenge in the flow synthesis of functional materials is avoiding deposition of product or reactants on the reaction channel wall. This can be a particular problem for materials like nanoparticles or polymers, where as the product grows it becomes less soluble, and has an increasing chance of deposition. The flow regime used for a flow reactor has a large impact on the extent of any fouling that may occur.

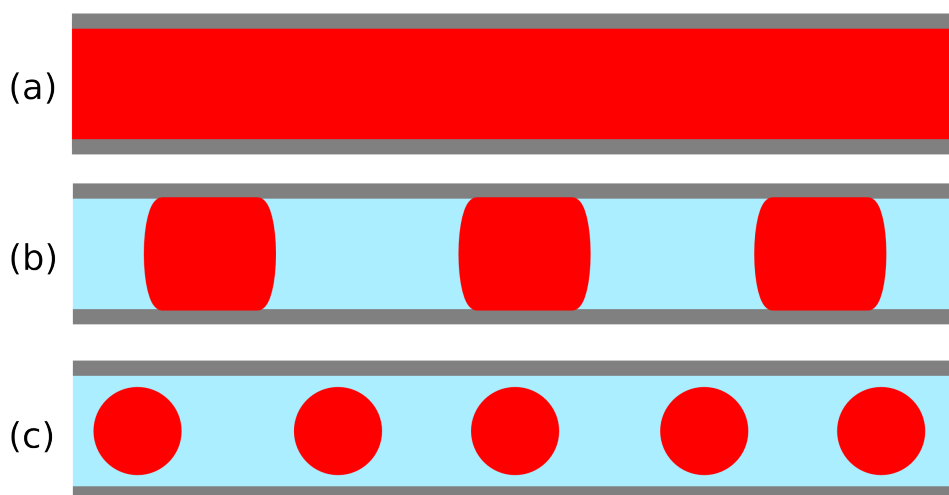


Figure 1.3: Diagram showing the different flow regimes commonly used in flow syntheses - (a) continuous flow where the reagent mixture is in contact with and completely wets the reaction channel walls, (b) slug flow where an immiscible carrier fluid is injected with the reagent mixture, causing it to form segmented slugs that still wet and are in contact with the channel walls, and (c) droplet flow where the carrier fluid preferentially wets the reaction channel walls, separating small droplets of reaction mixture from it.

The simplest flow regime is a continuous flow of reaction mixture, shown in Fig. 1.3a, where it completely fills and wets the walls of the reaction channel. Despite its simplicity, this is not always the best choice. Because the reaction mixture touches the walls of the reaction channel, there can be deposition and build-up of any insoluble or partially insoluble material. Build up of material will change the conditions inside the reaction channel and eventually block the reactor. Furthermore, contact between the flowing reaction mixture and the channel walls increases drag, leading to a parabolic distribution of flow velocities across the channel width and hence a spread of residence times.

Despite the high chance of fouling, continuous flow reactors have been widely used to synthesise both nanoparticles and polymers. For example, the flow syntheses of conjugated polymers by Seyler *et al*^{16,18} discussed above were carried out in continuous flow reactors. Similarly, continuous flow syntheses have been reported for a range of different nanoparticles^{25,30,31,34,47}. However, fouling of reactors is a well-documented problem in continuous flow^{22,49–53}, and where not explicitly mentioned in the literature it is most likely that syntheses were not run long enough to see the effects of material deposition, or were run at low enough concentrations to avoid it. The latter approach may be suitable for lab research but is not an option for industrial production, where high throughput is needed.

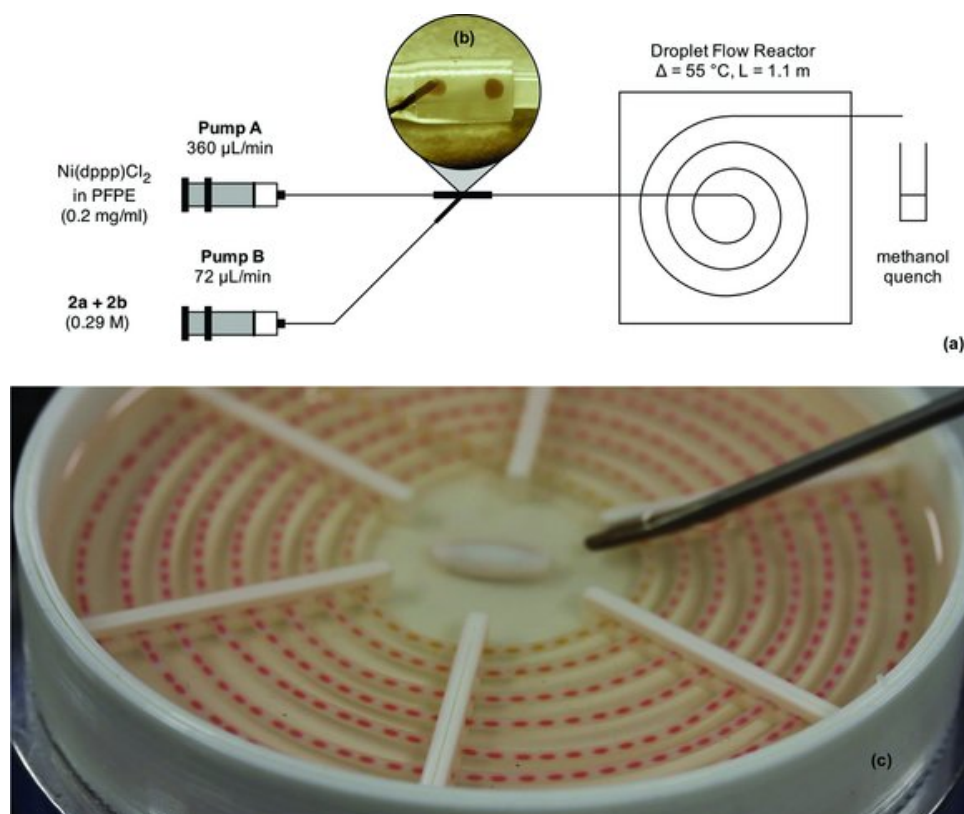


Figure 1.4: Diagram showing (a) a schematic of the synthesis of P3HT in a tube-based reactor, (b) droplet generation, and (c) a photograph the flow of droplets of reaction mixture through the reaction channel formed by PTFE tubing. Reproduced with permission from Bannock *et al*¹⁹.

To avoid these problems, a “segmented flow” can be used: an inert carrier fluid, like a gas or immiscible liquid, is co-injected with the reaction mixture into the reaction channel. If the reagent flow rate is high enough, or the reaction mixture preferentially wets the channel walls, a segmented flow is achieved where slugs of reaction mixture in contact with the channel walls are separated from each other by the inert carrier, as shown in Fig. 1.3b. The formation of well-separated slugs prevents the parabolic flow effect seen in continuous flow systems. Yet, the reaction mixture is still in contact with the walls of the reaction channel, and so the possibility of material deposition and fouling remains.

Using a carrier fluid that preferentially wets the reaction channel walls, for example a fluorinated polymer with a PTFE reaction channel, can allow discrete droplets of reaction mixture to form. In this droplet flow regime, shown in Fig. 1.3c, the droplets are separated from the reactor wall by a barrier of carrier fluid, preventing any material deposition and therefore fouling from occurring. However, the introduction of a carrier fluid reduces the maximum achievable throughput of the reactor, as some of the reaction channel is always occupied by the inert carrier.

Both slug and droplet flow syntheses have been reported for functional materials. A wide range of metal^{22,28,29,32,49,54,55}, metal oxide^{22,39,41,56}, and semiconducting nanoparticles^{22,51,57,58} have been synthesised in segmented flow reactors, with examples of control over nanoparticle size²⁹ and shape²⁸, as well as large scale syntheses over a long period of time^{24,49}. In terms of polymer synthesis, droplet-flow has been successfully used to synthesise high molecular-weight P3HT⁴³, avoiding problems with increasing viscosity as the product forms as well as the aforementioned product deposition. It has also been used to tune the stoichiometry (ratio between monomers) of a conjugated random copolymer⁴². An example of a droplet-flow reactor developed by Bannock *et al*¹⁹ for the synthesis of P3HT is shown in Fig. 1.4, showing the formation of droplets of reaction mixture in a carrier fluid using a bespoke droplet-generator, as well as the procession of these droplets through a reaction channel.

Despite their advantages, given the added complication and reduction in production rate, it is not always preferable to use a droplet or slug flow regime over a continuous one. If there are no signs of fouling (which there should not be if all reactants and products are soluble) and parabolic flow velocities can be tolerated, then the added complexity of using a carrier fluid should be avoided.

1.3 Automation and Optimisations

Another advantage of flow chemistry is the relative ease with which a flow reactor can be automated. Automation is gained by interfacing devices controlling reaction conditions, like the flow rates of reagents and temperature of the reaction channel, with a computer. This allows for even more precise control of reaction conditions, as well as improvements

to reproducibility, because the synthetic procedure is encoded in a set of commands for the computer that will be executed in the same way every time. This also increases efficiency by reducing the time and effort demanded of the chemist. These improvements in efficiency make experiments with a large number of data points, such as parameter screening and optimisations, more accessible.

By incorporating in-line analysis, flow reactors can further save a chemist time. Automatic product analysis not only removes the need to carry it out by hand, but the real-time analytical information can be fed into algorithms that then change the reaction conditions based on that information. Automated flow systems are naturally a subset of all flow syntheses, and as such there are fewer examples in the literature. Despite this, a number of important contributions have been made, with some focusing on functional materials synthesis.

1.3.1 In-line Analysis

In-line flow analysis means any analytic method that has been attached to the outflow of a flow reactor, in order to analyse the product mixture that has formed, as shown by the schematic in Fig. 1.5. A number of different spectroscopic and chromatographic techniques have been adapted to flow, with most common methods of product characterisation being reported^{59,60}.

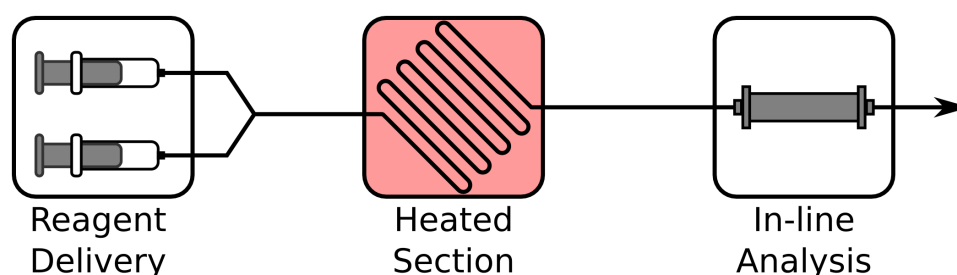


Figure 1.5: A schematic diagram of a flow reactor with in-line analysis, showing reagent delivery to a heated reaction channel, followed by in-line analysis of the product at the outflow of the reaction channel.

Many spectroscopic techniques have been applied to in-line analysis, including UV-visible^{61,62}, infra-red^{63–67}, and near IR⁶⁸ absorption spectroscopy and Raman^{69–71} spectroscopy for a range of synthetic targets. Nuclear magnetic resonance (NMR) spectroscopy has also been used for in-line analysis^{72–76}, with the recent development of bench-top NMR machines making it more accessible. In terms of functional materials synthesis, fluorescent emission spectroscopy has been used in-line as a way of estimating the size of quantum dot nanoparticles produced in droplet flow⁷⁷. Similarly, UV-vis absorption has been used to monitor the wavelength of longitudinal surface plasmon resonance (LSPR) in gold nanoparticles synthesised in continuous flow, in combination with in-line dynamic light scattering to estimate particle sizes⁶¹.

Chromatographic techniques have also been adapted to flow. These use the relative affinity of analytes to a stationary phase to separate the components of a sample by how long they take to pass through that stationary phase. Gas and liquid chromatography have been used to measure the product composition in flow during a Heck reaction⁷⁸, the methylation of 1-pentanol⁷⁹, and the optimisation of the synthesis of EGFR kinase inhibitor AZD9291⁸⁰, as well as other small molecule syntheses^{81–84}. Mass spectrometry has also been used in the flow synthesis of small molecules^{85–87}, for example to monitor the amidation of methyl nicotinate⁸⁷ in flow.

The wide range of analytical techniques described above demonstrate that there are many options available for in-line analysis when developing an automated flow synthesis. While there can be multiple different suitable techniques, the most suitable depends on the requirements of the synthesis as well as the nature of the species being analysed. The choice of analytical technique can also affect the design of the flow system. A distinction is sometimes made between ‘in-line’ and ‘on-line’ configurations of analytical systems in flow reactors⁸⁶. The former describes a system where all of the product is analysed as it exits the reactor, while in the latter representative samples of the reactor product are analysed. Naturally, in-line analysis provides more detailed real-time information about a flow synthesis than on-line analysis can. Furthermore, the lag time between the product leaving the reaction channel and being analysed is typically shorter. However, not all analytical techniques can be easily adapted to an in-line configuration, and so can only be used for on-line analysis. Spectroscopic techniques are often more naturally adapted to in-line systems, while chromatographic techniques are usually limited to on-line systems because of the longer times needed for sample analysis. In this thesis, ‘in-line’ analysis is used in the broader sense to describe any method of product analysis attached to the outflow of a flow reactor, and details of the specific system configuration given where necessary.

Of the techniques described above, spectroscopy - such as fluorescent emission and visible light absorption - appears to be the most used for analysing the flow synthesis of functional materials in-line. There have been no reported uses of in-line chromatographic analysis in the flow synthesis of functional materials. This could be because chromatographic techniques are unsuitable, due to the relatively large sizes of materials like nanoparticles and polymers compared to the small pore sizes of the stationary phases used in these methods. However, size-exclusion chromatography, a type of liquid chromatography, is already a common method for determining the molecular weight and molecular weight distribution of polymers. Furthermore, in-line chromatography has been widely used in the flow synthesis of small molecules, leaving little reason why it could not be used in the flow synthesis of smaller functional materials such as dye molecules.

Similarly, while mass spectrometry and NMR are regularly used for characterisation of materials in the laboratory, neither have been used in-line for the analysis of functional

materials produced in flow. This may be because many functional materials have structure dependent optical properties - in which case it would make sense to use some kind of optical spectroscopy to track how these properties change with reaction conditions. Another reason may be the relative difficulty of incorporating these two techniques into flow reactors, as well as the expense of the equipment involved. Again, however, both methods have been used for in-line analysis in the flow synthesis of small molecules, suggesting that they could be used for functional materials if necessary.

Despite this, a wide range of analytical techniques have been incorporated into flow reactors for in-line analysis. Therefore, there are many options for characterising functional materials produced in automated flow reactors. The lack of reported functional materials syntheses in automated flow reactors with in-line product analysis means that there is scope for adapting more of these syntheses, and bringing the benefits of in-line analysis to the tuning of novel functional material properties.

1.3.2 Self-Optimisation

One of the big advantages of using automated flow reactors for functional materials synthesis is the ability to use them for self-optimisations. Real time information about the reactor product can be analysed by a computer and fed into an optimisation algorithm that decides which reaction conditions to try next. This creates a feedback loop in which the computer sets some initial reaction conditions, analyses the product, and then feeds this information into the optimisation algorithm, which decides the next reaction conditions and sets them on the reactor. A simple schematic of this process can be seen in Fig. 1.6. This loop will continue until some user-defined condition is met, e.g. when some property of the reaction product, such as yield, has been reached an acceptable value.

There has been some notable progress in the development of self-optimising flow reactors in the last decade. Many of these focus on the optimisation of example reactions that are important for the synthesis of small molecules, or important intermediates. For example, one of the earliest examples was reported by McMullen *et al* in 2010⁸¹, where they carried out an automated optimisation of the yield of the Knoevenagel condensation of *p*-anisaldehyde and malonitrile. Since then, self-optimising flow reactors have been reported for optimising the yield of a number of other targets such as the Heck reaction⁷⁸, the alkylation of 1,2-diaminocyclohexane⁸², the methylation of alcohols⁷⁹, and the final stage of the synthesis of EGFR kinase inhibitor AZD9291 (an anti-cancer drug sold by AstraZeneca)⁸⁰. More complex optimisation targets than just the reaction yield have also been used, with Jumbam *et al*⁸⁸ reporting the optimisation of both production rate and yield of methylated ethers, while Moore *et al*⁸⁹ targeted the production rate and conversion efficiency of the Paal-Knorr reaction.

Despite this activity, there has been just one reported self-optimisation of a functional

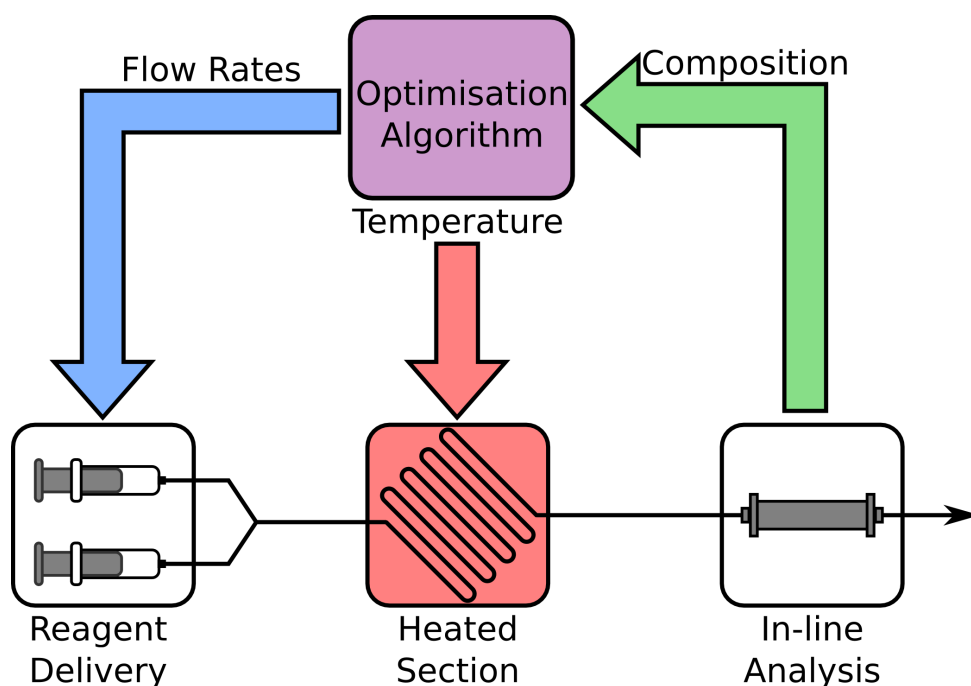


Figure 1.6: A simple diagram of an automated flow reactor for self-optimisation, showing reagent delivery to a heated reaction channel, followed by in-line analysis of the product.

Information about the product composition from the in-line analysis is fed into an optimisation algorithm, which creates a feedback-loop by controlling the flow rates and temperature of the reactor.

material. This was the use of an automated optimisation to tune the emission wavelength of chalcogenide quantum dots by Krishnadasan *et al* in 2007²¹, one of the earliest reports of a self-optimising flow reactor. By using an automated reactor that could vary the residence time, reagent ratio, and temperature, they were able to use the reaction conditions to control the size, and thereby emission wavelength, of the quantum dots. This allowed them to optimise the emission wavelength to a desired value with an intelligent optimisation algorithm. As well as tuning the emission wavelength, the same group also used a self-optimising reactor to maximise the emission intensity at a particular wavelength⁹⁰. Clearly there is room for the development of more self-optimising flow reactors for the synthesis of a wider range of functional materials, especially as many have properties that are tunable by varying reaction conditions.

Figure of merit

In any optimisation, a figure of merit is needed. This is the quantity or property that will be optimised. Often, these are formulated as minimisations, which means that in the examples above where a reaction yield was being optimised, the figure of merit might be set equal to negative of the reaction yield.

For a chemical synthesis, it is often the yield of a product that is the most important thing to optimise. However, there are also other quantities that may be of interest.

Examples mentioned above include the conversion efficiency from a starting material and the production rate of an end product. In terms of functional material synthesis, these targets might be properties of the materials like the emission wavelength of quantum dots or the molecular weight or regioregularity of a polymer, or they might be properties of the reaction like the relative yield of competing products.

When there is more than one quantity or property to be optimised, the optimisation becomes what is called a multi-objective optimisation. As most optimisation algorithms target a single figure of merit, it is necessary to transform these multiple objectives into one function that can be used as the figure of merit⁹¹⁻⁹³. The strategy employed so far in the literature of self-optimising flow reactors is to use a weighted combination of the optimisation targets, where the applied weights encode a relative preference for each target. For instance, Moore *et al*⁸⁹ used a weighted sum approach for their optimisation of conversion efficiency and production rate, while both Krishnadasan *et al*²¹ and Jumbam *et al*⁸⁸ used a weighted product of the properties that they were trying to optimise.

As in the examples mentioned above, optimising multiple criteria at once often involves balancing a trade-off - for instance, the reaction conditions that produce quantum dots that emit closest to a desired wavelength may have poor fluorescence intensity, requiring a compromise between getting the exact emission wavelength and high intensities. However, choosing appropriate weighting coefficients that capture what the chemist thinks is the best trade-off can be a difficult endeavour. Often, extensive experimentation is needed to select weights that give a sensible balance between the optimisation targets, and even then there is no guarantee that the outcome will match up to what the chemist requires.

If self-optimising reactors are to be used more, especially in the field of functional materials synthesis where there are often multiple properties that must be optimised, then a more intuitive method needs to be developed that lets the chemist encode their requirements into a figure of merit without prior experimentation or complicated mathematical manipulations.

Optimisation Algorithms

As well as needing to choose a figure of merit, it is necessary to select an algorithm to carry out a chemical optimisation. Generally, optimisation algorithms can be split into global and local optimisations, with the former able to find the overall optimum if allowed to run long enough, while the latter may not be able to find the overall optimum regardless of how long they run for. Regardless of type, though, all optimisation algorithms proceed by evaluating an objective function - the figure of merit described above - at an initial set of input parameters. They then use a set of rules and calculations to identify a new set of input parameters that may produce a more favourable figure of merit, in the usual case of minimisations this means a lower figure of merit than the ones before. After evaluating the objective function at the new set of input parameters, the new figure of merit information

is fed back through to identify another set of input parameters to further optimise the objective function. The algorithm typically loops around in this way until predefined termination criteria are met, such as a figure of merit falling below a certain threshold or no change in the figure of merit for a particular number of objective function evaluations.

Objective functions can be complex, with several minima. These minima may be local, where they are lower than all the surrounding points but not the overall minimum of the function, or global minimum. However, optimisation algorithms do not have information about the value of the objective function at all combinations of input parameters, what is called the parameter space. As such, they can wrongly identify a local minimum as the overall minimum for an objective function.

Local optimisation algorithms are satisfied with this, effectively treating the problem of knowing the global minimum as too costly, and instead finding the minimum within a particular subset of the parameter space. Many local optimisations are what are called gradient-based optimisation algorithms. As well as evaluating the objective function, these calculate or estimate the gradient of the objective function to determine what direction the minimum is likely to be found in. However, these are not usually preferred for chemical optimisations in flow reactors.

A chemical optimisation can be thought of as a black-box optimisation - the underlying function that relates the input parameters to the figure of merit is usually unknown, and therefore the objective function has to be evaluated directly. As such, in a gradient-based optimisation measurements need to be taken for gradient estimation, on top of those required for searching for the minimum. Yet, each evaluation of the objective function is costly in terms of both time and materials for a chemical optimisation. While the use of automated flow reactors reduces this cost somewhat, it is still desirable that the optimisation is carried out in as few measurements as possible.

Despite this, there is an example of a gradient-based optimisation algorithm used in a self-optimising flow reactor. In their paper from 2015, Reizman and Jensen used a gradient-based algorithm to optimise the yield in the alkylation of 1,2-diaminocyclohexane⁸², with the reaction solvent as one of the input parameters. They were able to do this by first carrying out three initial design-of-experiment (DoE) steps before the gradient-based optimisation. By using the initial DoE experiments, they were able to not only select the best solvent to use in the final optimisation, but also fit response surfaces to the synthesis in each solvent. The form of these response surfaces then made the gradient available for the optimisation. While this is an impressive display of how a well-designed optimisation can be used to optimise a particularly difficult problem, the use of the initial DoE steps to get the gradient information required the authors to carry out a significant number of extra experiments - just one of their DoE steps, the third one, resulted in an extra 27 experiments.

For these reasons, direct-search local optimisation algorithms (i.e. algorithms that

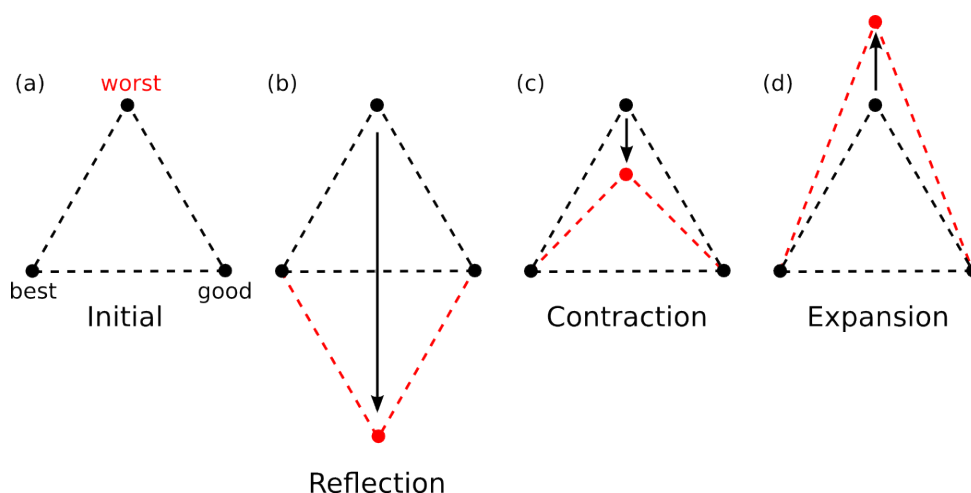


Figure 1.7: Diagram of (a) the initial triangle chosen during a 2D simplex optimisation with the points classified as best, good, and worst by the value of the figure of merit, alongside the possible ways of generating a new point: (b) by reflecting through the plane formed by the best and good points, (c) contracting the triangle, or (d) expanding the triangle.

do not require derivative calculations) are generally preferred for optimisations in flow. In particular, variants of the Simplex algorithm were used in a number of the self-optimisations mentioned above^{72,78,79,81,83,84,88,94}. The Simplex algorithm proceeds by first evaluating the objective function at points that form the vertices of a simplex in the parameter space. A simplex is a special shape that has $n + 1$ vertices, where n is the number of dimensions. For two input parameters, i.e. a two dimensional problem, a simplex is a triangle. After evaluating these points, the algorithm reshapes the simplex by replacing the vertex with the worst figure of merit by a function evaluation at a new point. This new point is chosen by reflecting the worst vertex through the line or plane formed by the other, better vertices. In the Nelder-Mead Simplex algorithm, a commonly used variant, the new point can also be chosen by expanding or contracting the simplex in the direction of the worst vertex, This helps to prevent the algorithm from getting stuck when the same vertex is identified as the worst after the reflection step, which would cause the simplex to flip back and forth. An illustration of these processes can be seen in Fig. 1.7. Although this has the advantage of directly searching the parameter space, and therefore typically needs fewer function evaluations than gradient based searching, it is still a local search method, and therefore does not guarantee that it will find the global optimum of a chemical synthesis.

Global optimisations, on the other hand, attempt to find the global minimum of the objective function. Some powerful global optimisation algorithms, like genetic algorithms and simulated annealing, achieve this by running the optimisation a large number of times with a change in parameters or starting point. This makes them unfavourable for chemical optimisations, where there is a need to keep the number of experiments required to a minimum. One global optimisation algorithm that has been used in chemical optimisations

is the SNOBFit algorithm⁹⁵. Along with variants of the Simplex algorithm, SNOBFit is one of the most widely used algorithms used in the literature on self-optimisations in flow reactors^{21,80,81,87,96}.

By using a combination of direct search in unexplored parts of the parameter space, and local fits around previously identified best points, the algorithm balances searching areas of the parameter space that have been found to have a low objective function value with searching parts of it that have not been explored yet, and may potentially have a better minimum in them. The algorithm proceeds by first evaluating the objective function at a number of points selected randomly from within the parameter space. It then splits the parameter space into boxes, so that each box contains one point that has been evaluated. Local fits are made around the points, with a full quadratic fit being used for the point identified with the lowest objective function value. Points that are expected to further minimise these fits are then recommended by the SNOBFit algorithm to be evaluated next. The size of the boxes around each point are used to determine where to search globally in parts of the parameter space that have not yet been explored. After the next set of points are evaluated, the parameter space is further partitioned into boxes containing just one point, and the local fits are refined with the new information. A demonstration of this process for an unknown function can be seen in Fig. 1.8.

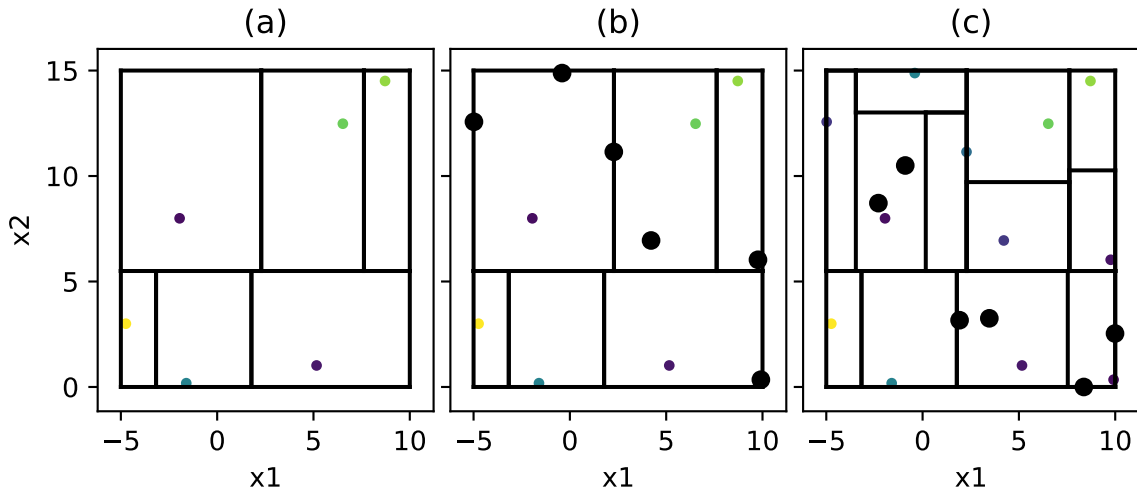


Figure 1.8: A series of scatter plots illustrating how points are recommended by the SNOBFit algorithm: (a) random initial points are evaluated in the first step and the parameter space is divided into boxes containing one point each, (b) the next set of points to evaluate are recommended in boxes with large amounts of unexplored space, (c) these points are evaluated, the parameter space further divided into boxes, and the next set of points are recommended as a mixture of points that may further minimise the current best points and points in boxes with large unexplored areas.

By sampling a mixture of expected minimisers from local fits and points chosen randomly from unexplored parts of the parameter space, SNOBFit avoids getting ‘trapped’ in local minima, and so will eventually converge on the global minimum if left to run long

enough. As such, SNOBFit can be classed as a global optimisation algorithm.

As well as being more likely to find the global minimum, SNOBFit has a number of properties that make it a good choice for chemical optimisations. Rather than recommending and then evaluating points sequentially, SNOBFit recommends points in batches of a user-determined size, allowing multiple syntheses to be carried out in parallel or in a more time efficient manner. It also uses a number of methods, such as stochastic fitting methods in the local fits, that make it tolerant to noise in the objective function, something that is almost unavoidable in a chemical synthesis.

Although it could be argued that a global optimisation algorithm is the most suitable for optimising a chemical syntheses, a recent paper by Tibbets *et al*⁹⁶ suggests that the response surface, a mapping of the underlying function, of most chemical processes is normally monomodal. In such circumstances, this would suggest that regardless of the algorithm used, as long as it is of a certain quality, an optimisation of a chemical synthesis will usually find the global minimum in relatively few objective function evaluations. However, some of the properties of SNOBFit mentioned above, such as noise tolerance and the ability to run parallel experiments, arguably make it a more suitable choice for self-optimising flow reactors.

Tibbets *et al* also noted, however, that the response surface of processes that balance competing properties can be more complex than normal chemical processes. This is because the interactions between the optimisation targets can create multiple local minima and maxima. In these cases the choice of optimisation algorithm is important, and one that is more likely to find the global minimum, like SNOBFit, would be preferable.

1.4 Conclusions

From the above discussion of the literature concerning the flow synthesis of functional materials, it can be seen that there has been a healthy focus on developing flow reactors for functional material synthesis over the past couple of decades⁹⁷, with a number of different solutions to some common problems. However, there are fewer examples of automated reactors using in-line analysis applied to this area, and fewer examples still of self-optimising flow reactors used to optimise the properties of functional materials.

Given the links between the structure and the properties of a functional material, and therefore between the synthetic conditions and properties of that material, self-optimising flow reactors are an enabling technology for the production of high-quality functional materials with tuned properties - such as emission wavelength, molecular weight, or purity. As such, there is room for a wide variety of functional materials beyond quantum dots to be synthesised in self-optimising reactors. Doing so would allow for different property problems to be solved, such as tuning the molecular weight of polymers or selecting one of a number of competing products.

Further to this, although a number of different optimisation algorithms have been used in self-optimising flow reactors, they remain unintuitive to use for more complicated problems, especially when optimising a number of competing objectives at once. Therefore, there is a need for the development of more intuitive methods for specifying the requirements of a chemical optimisation. Doing so would allow chemists to carry out automatic optimisations of a number of properties of a functional material without much prior knowledge of how the synthesis responds to reaction conditions, and without involved mathematical manipulation of formulae.

To these ends, the work described in this thesis attempts to further flow reactor technology by developing a more intuitive method for carrying out chemical optimisations with multiple competing targets, and by applying flow reactors to new functional material syntheses. In doing so, this work builds on previous work within the group, notably the self-optimisation work by Krishnadasan *et al*²¹ and the development of a three-phase droplet flow reactor by Nightingale *et al*⁵⁷.

CHAPTER 2

Experimental Methods

This chapter contains details of the materials and methods used for the work described in this thesis. This includes details of both batch and flow synthetic procedures, as well as the development and construction of the flow reactors used.

2.1 General Materials

Fullerene C₆₀ was purchased from Solenne BV, NL, all other reagents were obtained from Sigma Aldrich, UK. All solvents used for analysis were chromatography grade, and obtained from VWR, UK. Nuclear magnetic resonance (NMR) spectroscopy was performed at room temperature on a Bruker 400 MHz machine, using deuterated chloroform (CDCl₃) to dissolve the samples.

For all flow reactors, unless otherwise stated, the reaction channel was made from polytetrafluoroethylene (PTFE) tubing (1 mm I.D., 2 mm O.D., Polyflon Technology). Components of the flow reactors were linked to the reaction channels using the same tubing and connected together with flange-less PTFE nuts and ETFE ferrules (Upchurch Scientific), unless otherwise stated.

2.2 Batch Synthesis of *o*-Xylenyl Fullerene Adducts

2.2.1 Sultine Synthesis

Sultine (1,4-dihydro-2,3-benzoxothiin 3-oxide) was synthesised following the procedure reported by Hoey and Dittmer⁹⁸, as used in the synthesis of *o*-xylenyl functionalised fullerene derivatives published by Kim *et al*⁹⁹.

Sodium hydroxymethansulfinate (1.01 g, 8.53 mmols) and tetra-*n*-butylammonium bromide (206 mg, 0.64 mmols) were added to a gently stirred solution of 1,2-bis(bromomethyl) benzene (563 mg, 2.13 mmols) in DMF (38 mL). The reaction mixture was then cooled to 0 °C and stirred for four hours. Following this, the reaction was diluted with water, and the product extracted with diethyl ether to remove the reaction solvent, DMF. After washing the organic layer three times with water, the diethyl ether was removed by rotary evaporation at 30 °C. This afforded the sultine product as a pale yellow oil, that had a similar NMR spectrum to those previously reported^{98,100} (see Fig. 2.1).

2.2.2 Sultine Purity Estimation

Due to the low temperature of decomposition of the sultine, it was not extensively purified. The only purification took the form of washing the reaction product with water and removing solvent by rotary evaporation. Therefore, the product was contaminated with residual starting material and solvent, and so the purity was estimated by ¹H NMR.

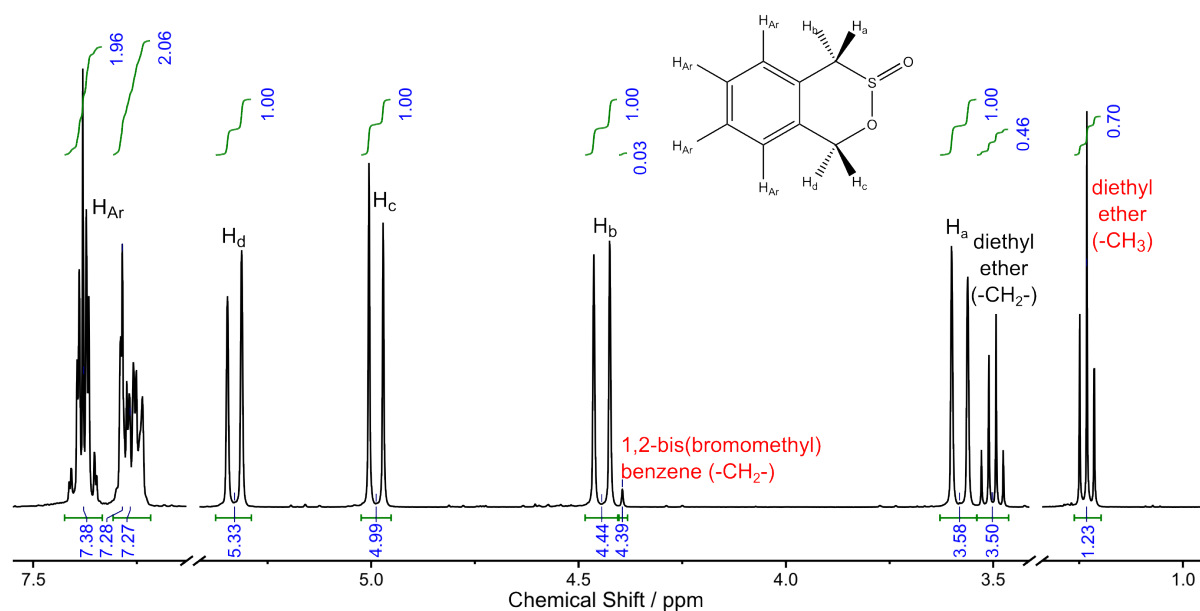


Figure 2.1: Example ^1H NMR spectrum of as-synthesised sultine, showing peak positions, their integrals, and their assignment to the different proton environments in sultine and the impurities. Impurity assignments have been indicated in red.

The main sources of impurity were the 1,2-bis(bromomethyl)benzene starting material, and residual diethyl ether. An example ^1H NMR spectrum is displayed in Fig. 2.1, showing the assignment of peaks to protons in the sultine molecule, as well as the significant impurity peaks. For sultine, there are two multiplets centred around 7.38 ppm and 7.27 ppm, corresponding to the four aromatic protons. There are then four doublets at 5.33 ppm, 4.99 ppm, 4.44 ppm, and 3.58 ppm due to the four methylene protons. The sultine aromatic multiplet region overlaps with another multiplet corresponding to the aromatic protons in 1,2-bis(bromomethyl)benzene, as well as a singlet from the NMR solvent CDCl_3 at 7.28 ppm. There is another singlet at 4.39 ppm due to the methylene protons in 1,2-bis(bromomethyl)benzene. The triplet at 1.23 ppm and the quartet at 3.50 ppm are due to the six methyl and four methylene protons in diethyl ether, respectively. A summary of these assignments can be found in Table 2.1.

To estimate the purity of a batch of sultine, the ratio of the integral of one of the sultine peaks to the sum of the integrals of the impurity peaks and the sultine peak was calculated (see Eqn. 2.1). One of the sultine doublet peaks was used in this calculation due to the overlap in the aromatic region; for consistency this was chosen to be the doublet at 5.33 ppm. The impurity peaks at 4.39 ppm and 3.50 ppm were also used in the calculation. As these all correspond to different numbers of protons in their respective molecules, each of the integrals was divided through by the corresponding number of protons they represent.

$$\text{Purity} = \frac{\int I_{\text{H}_a} dx}{\int I_{\text{H}_a} dx + (\int I_{\text{B}} dx / 4) + (\int I_{\text{D}} dx / 4)} \quad (2.1)$$

Molecule	Environment	Multiplicity	δ / ppm	Integral
sultine	Ar-H	multiplet	7.38 and 7.27	1.96 and 2.06
sultine	H _d	doublet	5.23	1.00
sultine	H _c	doublet	4.88	1.00
sultine	H _b	doublet	4.32	1.00
sultine	H _a	doublet	3.50	1.00
1,2-bis(bromomethyl)benzene	Ar-H	multiplet	7.27	-
1,2-bis(bromomethyl)benzene	-CH ₂ Br	singlet	4.39	0.03
diethyl ether	-CH ₂ -	quartet	3.50	0.46
diethyl ether	-CH ₃	triplet	1.23	0.70

Table 2.1: Chemical shifts and integrals of peaks assigned to the proton environments of sultine and the significant impurities in the ¹H NMR spectrum of the as-synthesised samples.

For all syntheses and stock solutions that used sultine, the purity was estimated in this way and solvent or reagent quantities were adjusted to give the required concentrations. Typically the purity lay in the range of 75 to 90 %.

2.2.3 Batch Synthesis of Fullerene Derivatives

A typical batch synthesis of *o*-xylenyl functionalised fullerene derivatives involved dissolving the as-synthesised sultine (319 mg, 1.9 mmol) in *o*-DCB (46 mL) and adding it to a flask containing fullerene C₆₀ (456 mg, 0.63 mmol), under argon. The reaction mixture was heated to 80 °C, and stirred at this temperature for 18 hrs. The reaction was then cooled to room temperature, before washing with water and removing the solvent by rotary evaporation at 80 °C to give a fine brown powder (typically ~800 mg). Analysis by a self-built HPLC system (described in Section 2.3.2) revealed this to be a mixture of fullerene derivatives. A representative chromatogram is shown in Fig. 2.2a, representing a composition of 0.5 % X₀, 33.3 % X₁, 49.9 % X₂, and 16.3 % X₃, where X₀ is C₆₀ and X₁, X₂, and X₃ are the singly, doubly, and triply functionalised derivatives.

The above procedure, using a molar ratio of sultine:C₆₀ of 3:1, resulted in a product mixture rich in X₂. This procedure was altered to target a product mixture rich in X₁ by using a molar ratio of sultine:C₆₀ of 1:1, and heating at 80 °C for five hours. A representative chromatogram is shown in Fig. 2.2b, corresponding to a composition of 23.5 % X₀, 72.6 % X₁, 3.8 % X₂, and 0.1 % X₃. For a product mixture rich in X₃, a molar ratio of sultine:C₆₀ of 5:1 was used, and heated at 150 °C. Again, a representative chromatogram is shown in Fig. 2.2c, corresponding to a composition of 0.3 % X₀, 18.1 % X₁, 53.4 % X₂, and 28.2 % X₃. The conditions used in batch to target the different

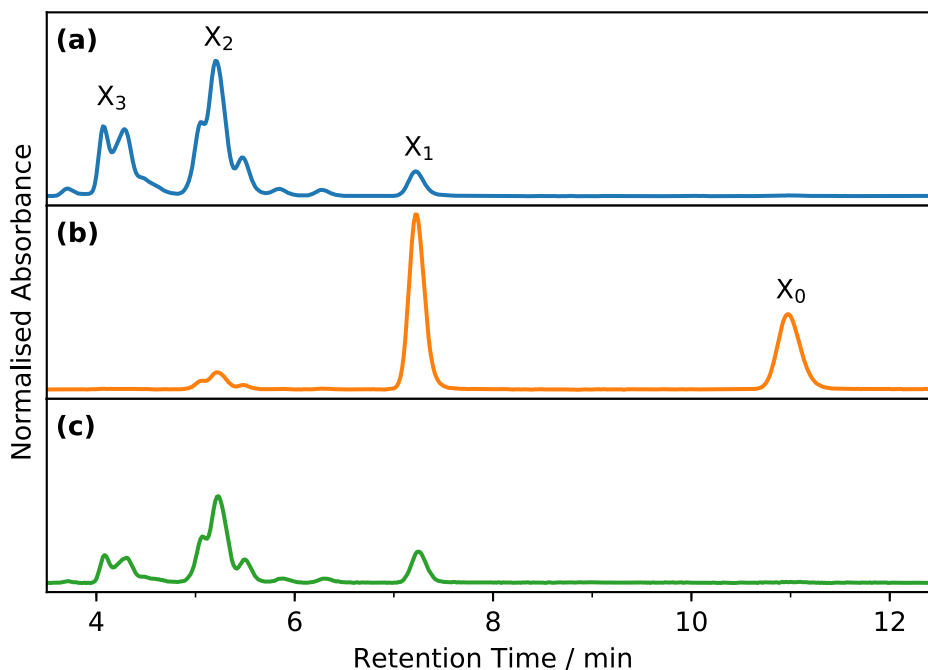


Figure 2.2: Chromatograms of *o*-xylenyl functionalised fullerene derivative mixtures synthesised in batch using (a) [Sultine]:[C₆₀] = 3:1, temperature = 80 °C, and reaction time = 18 hrs, (b) [Sultine]:[C₆₀] = 1:1, temperature = 80 °C, and reaction time = 5 hrs, and (c) [Sultine]:[C₆₀] = 5:1, temperature = 150 °C, and reaction time = 5 hrs.

derivatives were not optimised, but were instead derived from chemical intuition.

2.2.4 Separation of Fullerene Derivatives

Samples of each of the single *o*-xylenyl functionalised fullerene derivatives were separated from product mixtures synthesised in batch to provide standards for calibration of the self-built HPLC system, the process of which is described in Section 2.3.2. Separation was carried out by column chromatography on silica. Hexane was used as the eluent at first, and the polarity was increased after each fullerene derivative eluted using toluene. Purity was first confirmed using thin layer chromatography, and then qualitatively with the self-built HPLC system.

2.3 Construction of an Automated Flow Reactor

In constructing this continuous-flow system, two main components needed to be designed and built: a solid-state heater to control the temperature of the reaction channel, and a high performance liquid chromatography (HPLC) system for in-line analysis of the reactor product.

2.3.1 Development of a Solid-State Heater

A stable method of heating the reaction channel was required to ensure reproducible results from the flow reactor. Oil-baths are commonly used in synthesis labs to heat reactions because of their simplicity and versatility. However, they are cumbersome and can be complicated to use with tube-based flow reactors because the reaction channel must be held in a stable coil inside the bath. Therefore, a solid-state heater was designed so that the reaction tubing could be wound around it, providing heat and holding the reaction channel in place at once.

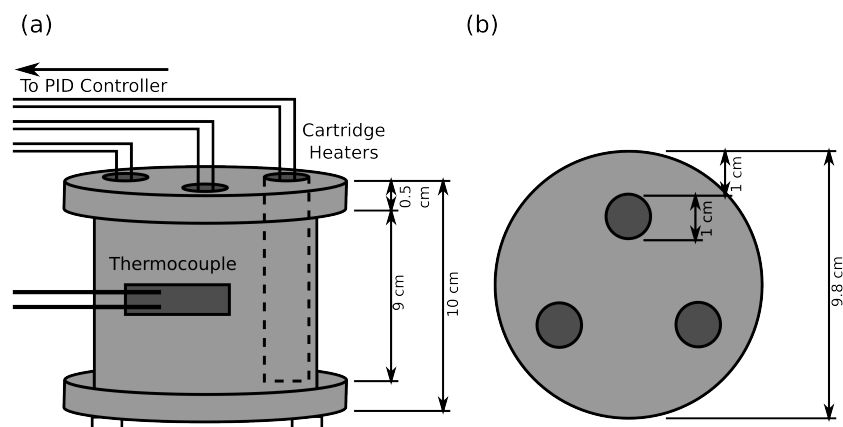


Figure 2.3: (a) Diagram of the solid state heater showing the aluminium block, the positioning of the K-type thermocouple, and the depth of the cartridge heaters (the dotted line), and (b) a plan view of the heater showing the configuration of the cartridge heaters.

The body of this solid state heater was formed from a cylindrical block of aluminium (9.8 cm diameter, 10 cm height), machined to have a lip (0.5 cm) at each end. A small hole was drilled through each lip to allow the reaction channel tubing to be threaded through and held in place on the heater. Three cartridge heaters were placed in holes drilled into the top of the heating block, and PTFE feet were attached to the bottom. A flag-shaped K-type thermocouple was attached to the side of the aluminium block to measure the surface temperature. A schematic of this is shown in Fig. 2.3.

The cartridge heaters were controlled by an Arduino microcontroller, to which the thermocouple was attached through a thermocouple amplifier chip (MAX6675, Adafruit). A PID control script, using the Arduino PID library (<https://playground.arduino.cc/Code/PIDLibrary>), allowed commands to be sent from the computer to read the temperature of the heater, control the set-point, and turn the heater on and off. The control unit was enclosed in a box that displayed the current heater temperature and set-point on an inset LCD display.

Using three cartridge heaters in the solid-state heater ensured that it could heat up rapidly. However, for randomised experiments or optimisations the heater would need to be able to both increase and decrease in temperature between consecutive measurements.

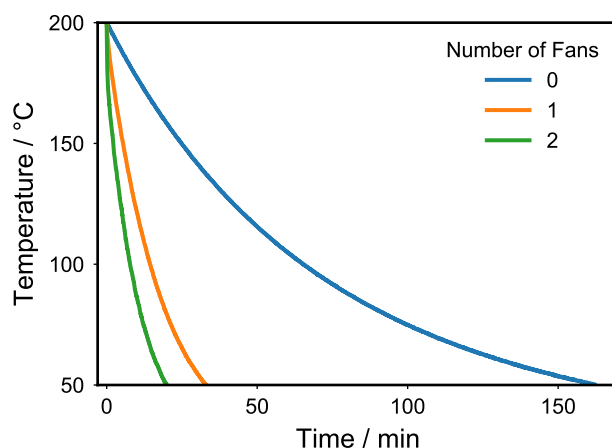


Figure 2.4: Plot showing the time taken for the heater to cool to 50 °C, with 0, 1, and 2 cooling fans attached.

To prevent experiments taking too long, it was necessary for the heater to be able to cool down within a reasonable time frame. The time that it took the heater to cool down was tested by heating it to 200 °C, turning it off, and timing how long it took to cool to 50 °C. This range was chosen as the appropriate range over which an optimisation of the synthesis of *o*-xylenyl functionalised C₆₀ derivatives could take place. Results for unassisted cooling can be seen as the blue line in Fig. 2.4, which shows that the heater took over 2.5 hours to cool completely to 50 °C. This was clearly impractical.

To cool the heater faster, cooling fans were added by building a box out of laser-cut perspex, with cut-out sections to accommodate the fans (see Fig. 2.5a). These fans were incorporated into the PID control system described above (see Fig. 2.5b). The time taken for the heater to cool from 200 °C to 50 °C was measured again, first with one fan cooling the heater, and then with two fans placed in a push-pull configuration. The resultant cooling curves can be seen in the orange and green curves of Fig. 2.4, which both display a marked improvement over the heater without fans. Using one fan, the heater reached 50 °C in 33 min, and using two fans it reached 50 °C in 20 min. Therefore, two fans were incorporated into the heating unit for the flow reactor (see Fig. 2.5a).

As well as being able to cool quickly, the heater needed to be able to heat up to the desired temperature rapidly with minimal overshoot, come to a stable temperature quickly, and then remain at that temperature with as little variability as possible. In order to measure these properties, the set-point of the heater was increased from 50 to 200 °C in 50 °C increments, waiting 30 min between each increase in the set-point. This test was performed twice: once without any cooling fans (Fig. 2.6a) and once with two cooling fans (Fig. 2.6b). As the fans were incorporated into the PID control loop, they turned on when the PID algorithm output a negative value and turned at a speed proportional to the magnitude of that value. For both configurations, the PID parameters were $K_p = 4$, $K_i = 14.8 \text{ s}^{-1}$, and $K_d = 0.36 \text{ s}$. These were chosen by tuning the heater with the

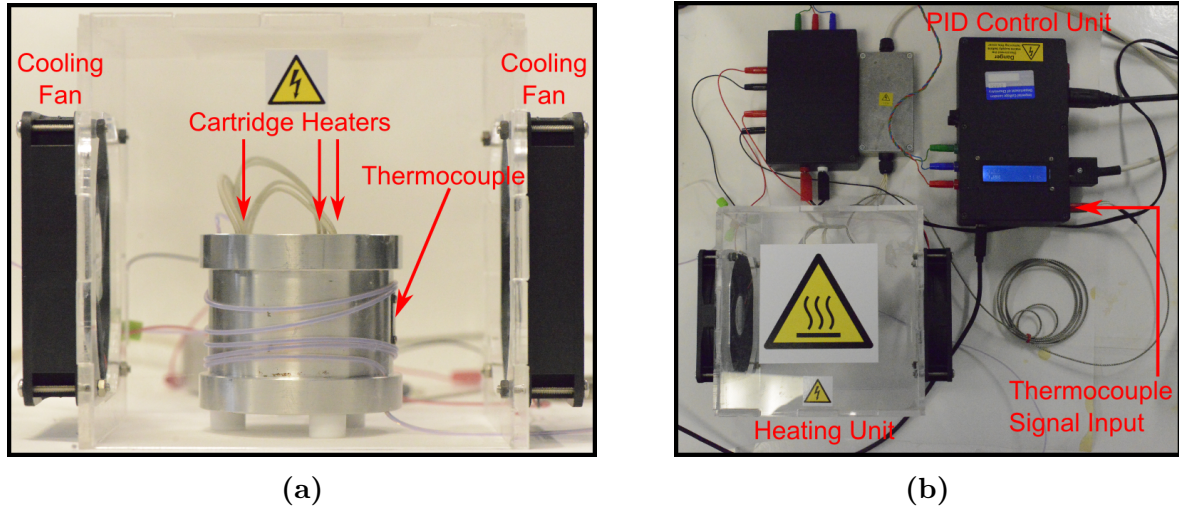


Figure 2.5: Photographs of the solid state heater from (a) the front, with the fans, cartridge heater, and thermocouple labelled, (b) the top, with the heating unit, PID control unit, and thermocouple input labelled.

auto-tune programme of a commercial PID controller.

As would be expected, both configurations of the heater had almost the same ramping time between temperatures, taking between 3 and 4 min for every increase in set-point. The biggest difference, however, was how far the temperature overshoot the set-points in each configuration. Without fans, the overshoot was large for all set-points: 11, 10, 9, and 8 °C for 50, 100, 150, and 200 °C respectively. Furthermore, it took a long time to get down to the desired temperature after these overshoots: when the set-point was 50 °C, the heater only just reached the desired temperature after 30 minutes. Using two cooling fans reduced the overshoot to 5, 5, 5, and 6 °C at 50, 100, 150, and 200 °C respectively. Therefore, as well as decreasing the cooling time, using two fans with the heater reduced the amount of overshoot when heating up to a temperature.

On the other hand, the cooling fans caused a large amount of noise in the temperature when it reached the desired temperature. In this heating test, the standard deviation of the temperature once it had reached the set-point was 0.7, 10.1, 10.1, and 9.9 °C at 50, 100, 150, and 200 °C. From Fig. 2.6b it can be seen that these high standard deviations were caused by an oscillatory ‘ringing’, where the heater rapidly alternated between heating and cooling modes. This also explains why the standard deviation was much lower at 50 °C - the rate of heat transfer to the surroundings was fast enough that the cooling fans were not activated. This suggests that, while the two cooling fans improved the behaviour of the solid-state heater, the parameters of the PID control algorithm needed to be tuned to provide the best performance.

The behaviour of a PID control system can be changed by altering the three PID parameters: the proportional constant K_p , the integral constant K_i , and the derivative constant K_d . These were systematically tuned by starting with a high proportional con-

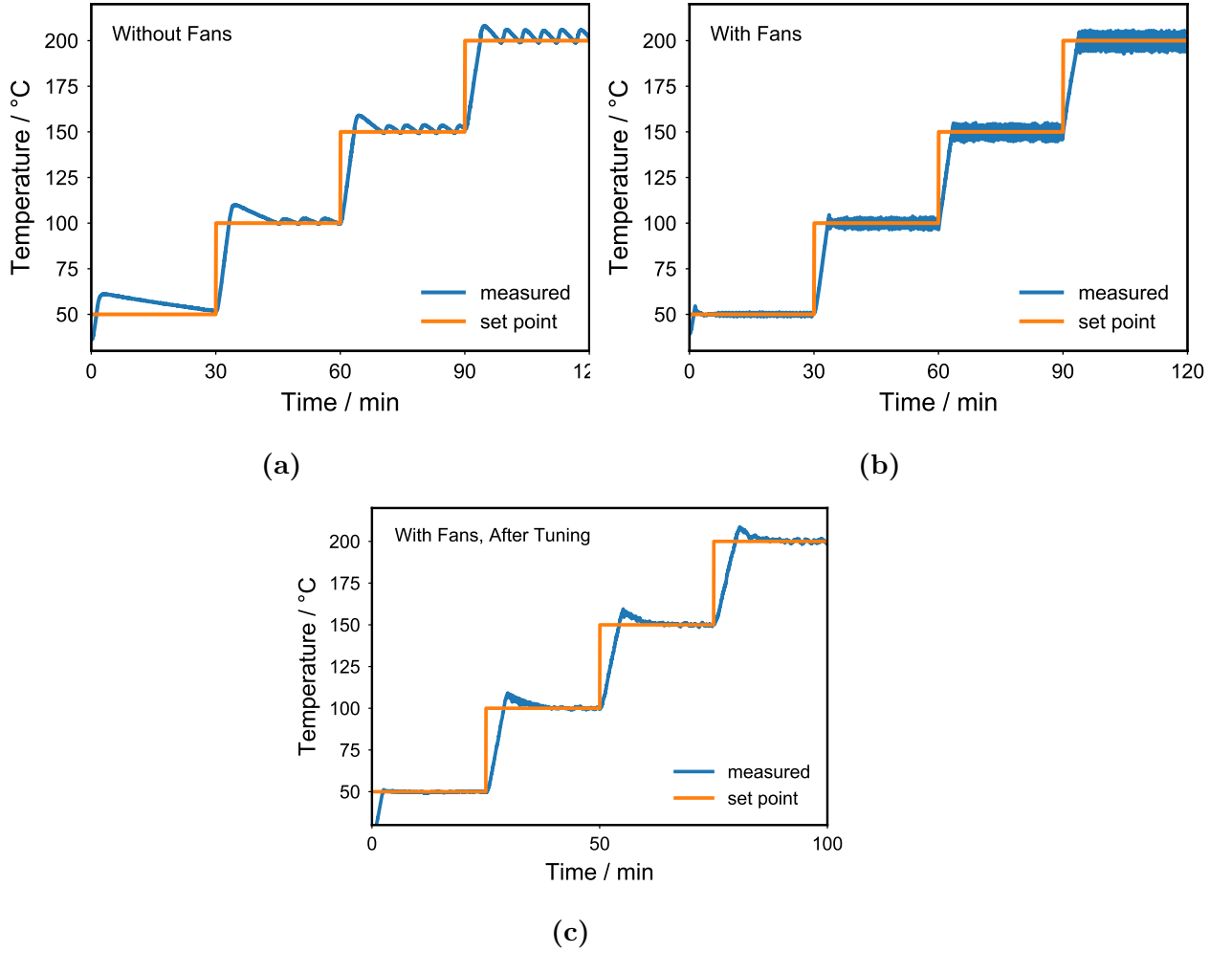


Figure 2.6: Measured temperature of the solid-state heater as the set-point was increased in 50 °C increments from 50 °C to 200 °C for the heater (a) without any cooling fans, (b) with two cooling fans, and (c) with two cooling fans after tuning the PID parameters.

stant, and the other constants set to zero. The integral constant was then increased until the overshoot was reduced, and then the derivative constant was increased until any oscillations were removed. The tuning was accepted when the standard deviation of the measured temperature and the mean deviation of the temperature from the set-point were both below 1 °C.

The parameters that achieved this desired behaviour were $K_p = 300$, $K_i = 20 \text{ s}^{-1}$, and $K_d = 0.6 \text{ s}$. The heating characteristics of the tuned heating-block were again tested by increasing the set-point of the heater from 50 °C to 200 °C in 50 °C increments and letting the heater remain at each temperature for 25 min (see Fig. 2.6c). From this it can be seen that although there was some overshoot as the heater ramped up to each set-point, the temperature settled to the desired set-point shortly after. The large oscillations in temperature seen for the untuned heater were no longer present, although some small oscillations could be seen as the set-point increased.

Fig. 2.7a shows the first 12 min of heating for each temperature. With a set-point

of 50 °C there was almost no overshoot and the temperature stabilised very quickly, but all other set-points took at least twice as long to heat to the desired temperature and exhibited some overshoot. The heater reached a stable temperature by 12 min for all set-points, and the final 13 min of heating in Fig. 2.7b shows the measured temperature was very stable and was almost exactly the same as the corresponding set-point.

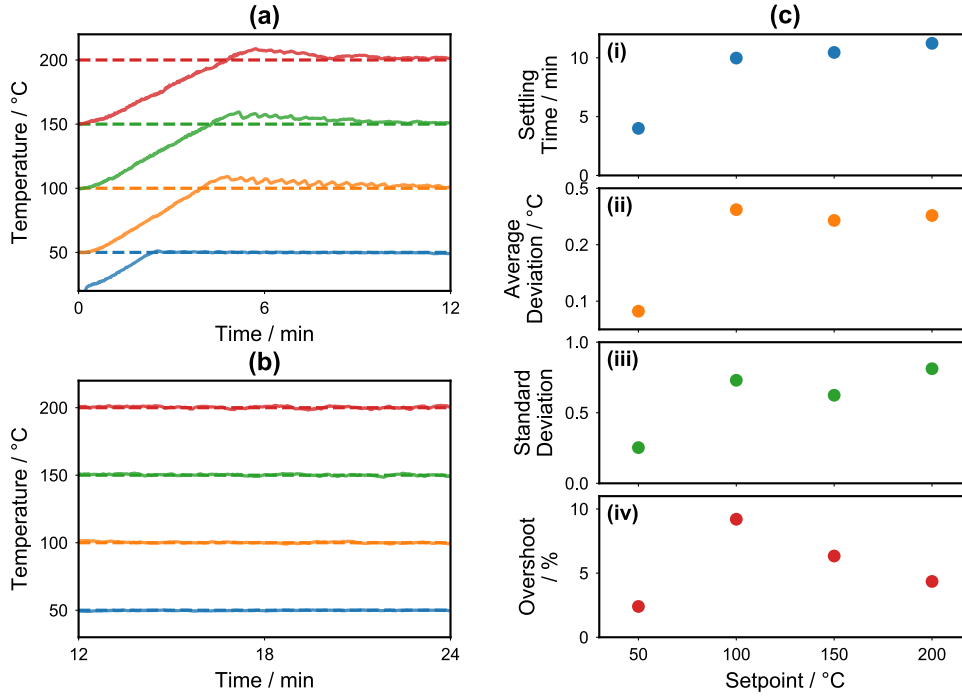


Figure 2.7: Temperature profile of heating-block at different setpoints (a) before reaching a stable temperature and (b) once reaching a stable temperature. (c) Important parameters of the temperature profile at each setpoint.

This can be seen in more detail in Figure 2.7c(i-iv), where the settling time, average deviation from the set-point, standard deviation of measured temperature, and overshoot are shown for each set-point. These plots show that the settling time increased with temperature, from 4.0 min at 50 °C to 10.0, 10.5, and 11.2 min at 100, 150, and 200 °C. Both the standard deviation of the measured temperature and the average deviation of the measured temperature from the set-point were lowest at 50 °C, at 0.25 and -0.15 °C respectively. These rose to 0.73 °C and 0.39 °C at 100 °C, before falling to 0.62 and 0.33 °C at 150 °C and then rising again to 0.81 and 0.36 °C at 200 °C. As a percentage of the set-point, the overshoot also reached a maximum at 100 °C, at 9.2 % compared to 2.4 %, 6.3 %, and 4.4 % for 50, 100, and 200 °C respectively. While the overshoot might be expected to increase as the set-point temperature increases, this can be explained by the fact that the fans attached to the heater turned on before the overshoot reached a maximum. As the rate of heat transfer to the surroundings is faster at higher temperature, the fans

cooled the heater faster at higher set-points, thereby slowing the increase in temperature faster and reducing the overshoot.

Overall, these tests indicated that the self-built solid-state heater provided a stable temperature that could both be increased and decreased relatively quickly, thereby giving good control over reaction temperature in the automated reactor.

2.3.2 HPLC System Development

A schematic of the high performance liquid chromatography (HPLC) system developed for in-line analysis of *o*-xylene functionalised fullerene derivatives can be seen in Fig. 2.8.

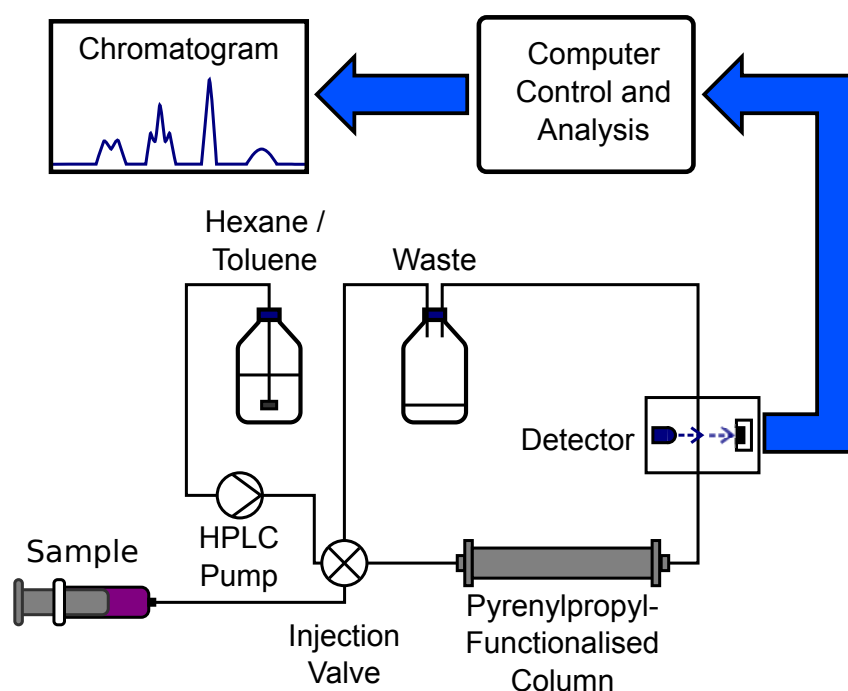


Figure 2.8: Diagram of the purpose-built High Performance Liquid Chromatography (HPLC) system

The system used a HPLC pump (Model 12-6, SSI) to pass the eluent solvent through an analytical column, packed with pyrenylpropyl-functionalised silica (BuckyPrep, COS-MOSIL). A remotely operated high-pressure six-port switching valve (Rheodyne MXP-7900) was used to inject samples into the system for analysis. A labelled photograph showing the arrangement of this equipment can be seen in Fig. 2.9a. The configuration of this injection valve is shown in Fig. 2.9b. In the ‘load’ position, a sample could be loaded into the sample coil (stainless steel, Upchurch Scientific, $V_{\text{coil}} = 5 \mu\text{L}$) while eluent solvent flowed directly into the column. In the ‘inject’ position, the eluent solvent was directed through the sample coil, carrying the sample into the column. Steel tubing was used to connect the high-pressure parts of the HPLC system, i.e. the pump to the switching valve, and the switching valve to the column.

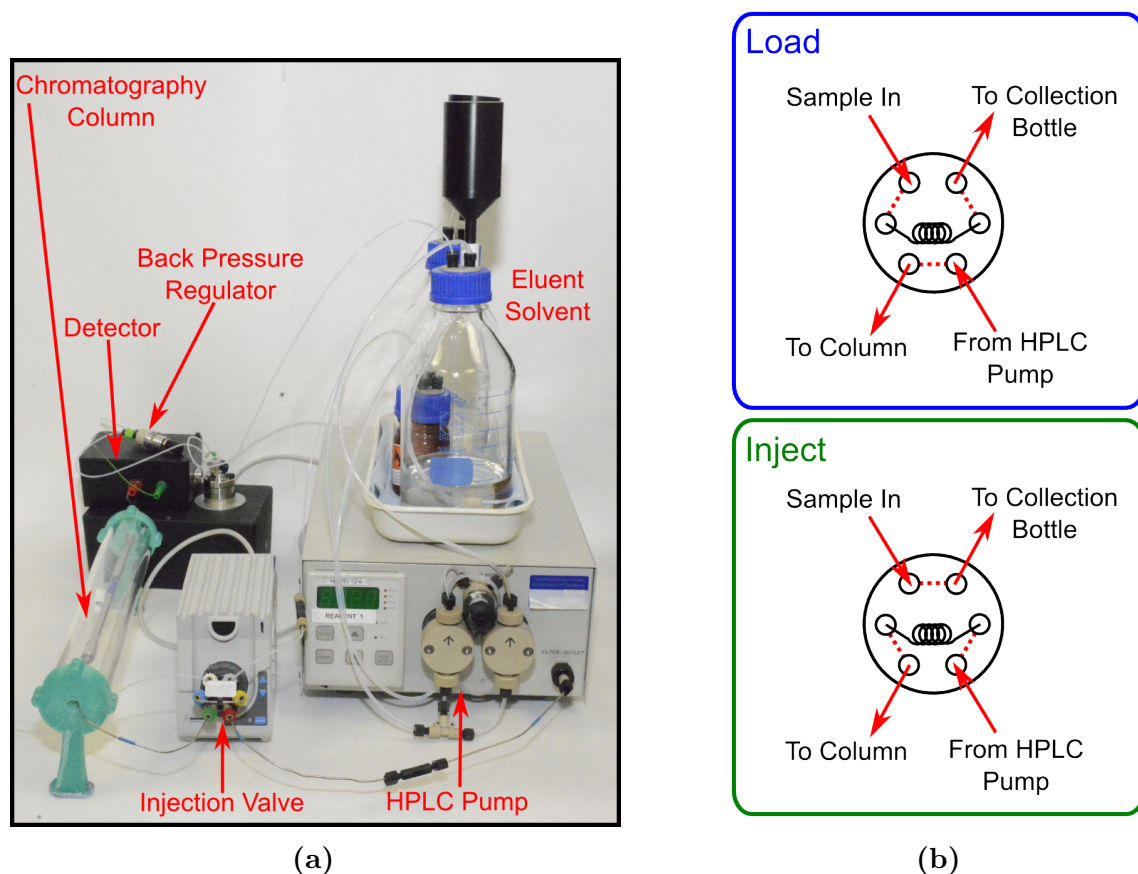


Figure 2.9: (a) A photograph of the self-built HPLC system, comprising a bottle containing eluent solvent, a HPLC pump, an injection valve, a chromatography column, a detector, and a back pressure regulator. (b) Schematic of the injection valve in the ‘load’ position, where the channels in the valve allow a sample to be loaded into the sample coil, and the ‘inject’ position, where the eluent solvent is routed through the sample coil to inject a sample into the column.

The optical transmission of the sample after the column was measured using a purpose-built analyser. The analyser was formed from a 390 nm light-emitting diode (LED) and an amplified silicon photodiode (OPT101, Texas Instruments) placed either side of transparent perfluoroalkoxy (PFA) tubing (0.50 mm I.D., 1/16 in O.D., Upchurch Scientific). A second amplified silicon photodiode was placed perpendicular to the LED to correct the signal for any fluctuations in the LED intensity, and the LED was powered through a constant current circuit to minimise any fluctuations. The optical components were held in place by a 3D-printed holder, and housed in a box to minimise external interference. The analyser components were then connected to a DAQ card (NI-6211, National Instruments) via a serial cable, which in turn was connected to a computer. A wire was also connected to the DAQ card from the injection valve, which sent a signal each time the valve changed position, thereby allowing signal acquisition to be triggered when a sample was injected into the column. Two fluidic ports were placed either side of the analyser box so that the outlet of the HPLC column could be connected.

A variable back-pressure regulator (JR-BPR 2, Vici Jour) was placed at the outlet of

the detector and used to control the total pressure in the HPLC system. An automated switching valve was placed after this to change between directing the outflow of the column to a collection bottle, or back into the eluent solvent reservoir. In doing so, eluent solvent could be circulated through the system when it was not being used for analysis, to maintain a stable isocratic/isobaric environment for liquid chromatography.

Once built, and before using the system for any analysis, eluent solvent was pumped through the system for 3 days until the pressure was constant at around 750 psi. The system was typically left running with just solvent pumping through it between analysing samples. If turned off, the system was allowed to re-stabilise before analysis was performed

Sample Analysis by HPLC

Sample analysis was computer controlled, using a MATLAB script. When started, the user was prompted to input the number of samples to be analysed, the expected maximum elution time, and an identifier for that set of analyses. For each sample, after allowing time for the user to load their sample into the sample coil, the script would turn the injection valve to the 'inject' position. After waiting sufficient time for ten volumes of the sample coil to be displaced, the injection valve would move back into the 'load' position, and the user could load their next sample if necessary.

Signal acquisition was triggered when the sample was injected. During acquisition, the detector LED was pulsed with a square wave at a frequency of 112 Hz, and the signals from each photodetector convolved with this same square wave as they were recorded, as a form of software lock-in amplification. This served to reduce the noise in the signal from external sources such as ambient light or electrical equipment. The signal from the photodetector opposite the LED was then corrected by the other photodetector, and recorded in a file for analysis later. Once analysis of all samples was finished, the detector light source was turned off to preserve the life of the UV bulb and a flow of solvent circulated through the column to keep the column primed for future measurements.

Chromatogram Processing

The signal measured by the detector was proportional to the intensity of light transmitted through the sample eluting from the column. A chromatogram is conventionally shown as the absorbance of the eluting sample over time, since at a given wavelength and elution time this is proportional to the concentration of the eluting species. Therefore, the signal needed to be converted into absorbance before the chromatograms could be analysed. Furthermore, changes in the system conditions could sometimes cause a slope in the baseline of the signal, which needed to be removed.

The measured signal was processed into a chromatogram using the following steps.

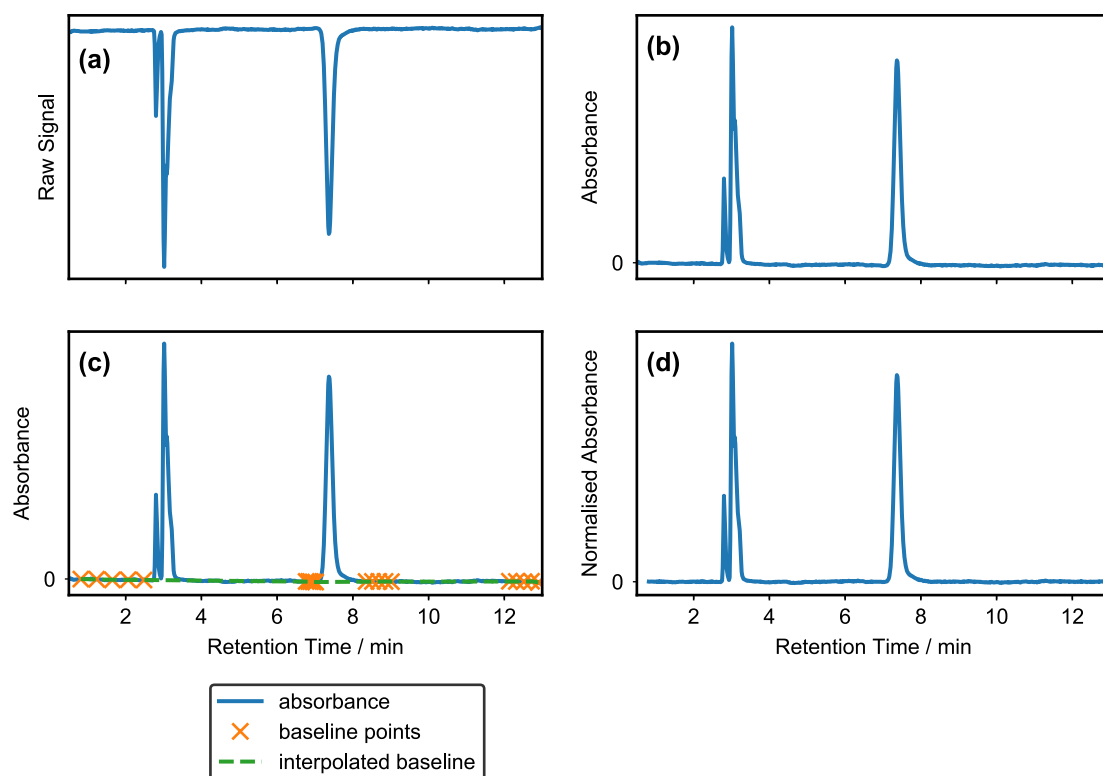


Figure 2.10: An illustration of how (a) the raw signal from the HPLC detector is processed by first (b) converting it to absorbance , (c) interpolating the baseline about fixed points with no analyte signal, and then (d) subtracting the baseline from the absorbance and normalising by the area of the solvent peak to get the final chromatogram.

The signal was first converted to absorbance using Eqn. 2.2:

$$A(t) = -\log_{10} \frac{I(t)}{I_0} \quad (2.2)$$

where the absorbance at a given time ($A(t)$) was calculated by using the ratio between the instantaneous measured signal (I) and the average reference signal (I_0) measured over the first 100 points when no absorbing species were eluting. The baseline signal was then estimated using a spline interpolation between points measured at intermediate times when there were no eluting species, and was then subtracted from the absorbance. Finally, the area of the whole chromatogram was normalised to the area of the solvent peak. The code for this process is provided in the Appendix, and the process is illustrated in Fig. 2.10. The chromatogram used in this illustration already has a fairly flat baseline, and so the baseline correction has a minimal effect. This is representative of the chromatograms acquired using the HPLC system, although some chromatograms did show sloping baselines, particularly if there was a change in system conditions that had not sufficiently stabilised before analysis. For the screening and optimisation studies reported later, the system was always allowed to stabilise first.

Sample Analysis by HPLC

A representative example of a chromatogram produced by analysing a mixture of fullerene derivatives with the self-built HPLC system can be seen in Fig. 2.11a. Each peak has been labelled with the species that it identifies. The first peak in the chromatogram was for the solvent that the sample was dissolved in, *o*-DCB, at a retention time of 3.0 min. After this, the fullerene derivatives eluted in reverse order of substitution, with each additional *o*-xylenyl group decreasing the affinity of the molecule to the column stationary phase. The peak for the triply-substituted derivative, X_3 , was seen first at 4.2 min, followed by X_2 at 5.4 min, X_1 at 7.5 min, and finally the unreacted X_0 at 11.5 min.

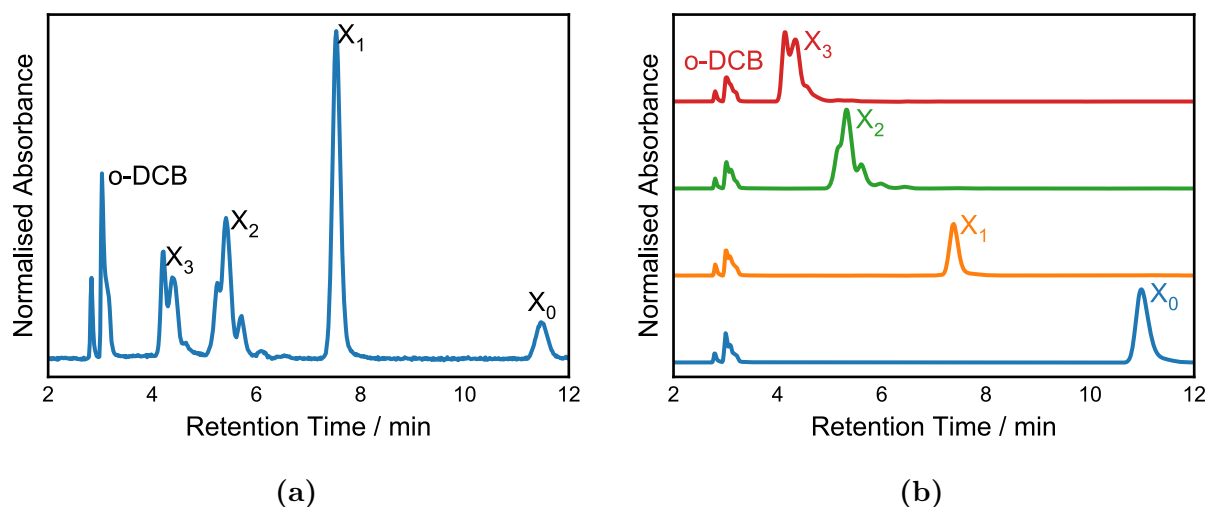


Figure 2.11: Example chromatograms acquired by the self-built HPLC system for (a) a typical product mixture and (b) pure samples of each of the *o*-xylenyl fullerene derivatives. The peaks have been labelled with the fullerene derivative that they are characteristic of - the typical retention times are 4.2 min for X_3 , 5.4 min for X_2 , 7.5 min for X_1 , and 11.5 min for X_0 .

The peaks for X_0 and X_1 are Gaussian-shaped as is expected for a single pure substance in a chromatogram. However, the peaks for X_2 and X_3 both look like clusters of overlapping peaks. This is because the *o*-xylenyl groups can attach in different configurations on the C_{60} cage, resulting in a mixture of regioisomers of X_2 and X_3 being formed during the synthesis¹⁰¹. These regioisomers have slightly different retention times, so appear as overlapping peaks in this chromatogram. Given how close these clusters of peaks were for X_3 and X_2 , there was a possibility that they overlapped. However, chromatograms of pure samples of each adduct confirm that there was no significant overlap between any of the adduct peaks, or clusters of peaks (Fig. 2.11b).

Eluent Solvent Composition Reproducibility

As well as causing sloping chromatogram baselines, changes in the eluent solvent composition or system pressure of a HPLC can lead to variation in the shape, height, and retention

time of peaks. While the system pressure could easily be controlled by the variable back-pressure regulator placed after the sample analyser, and was indicated by a display on the HPLC pump, changes in the composition of the eluent solvent were not as easily identified or corrected. The eluent solvent used here was a mixture of hexane and toluene mixed together volumetrically by hand. Although care was taken to do this as accurately as possible, this may have been a source of variation in the eluent solvent composition, and therefore in the chromatograms measured by the self-built HPLC system.

In order to find out if and how much of a problem this variation was, a test was run where samples of X_0 were measured using the HPLC system with different batches of pre-mixed eluent. The samples of X_0 were all taken from the same 1 mg/mL solution of X_0 dissolved in *o*-DCB. Six batches of eluent solvent were tested, and each measurement was repeated six times. The retention time, height, and area of the peaks were measured. The relative proportion of hexane and toluene in the eluent solvent was important for the separation of the different fullerene derivatives, as a higher fraction of toluene in the solvent would have caused all derivatives to elute faster from the column, decreasing their retention time. As such, a large variation in the eluent solvent composition would have caused a large variation in the fullerene derivative retention times. Similarly, changes in the retention time can also cause changes in the shape of peaks in the chromatogram, which would be reflected in a variation in the peak height. On the other hand, the areas of the peaks should not be affected by any changes in the peak shape.

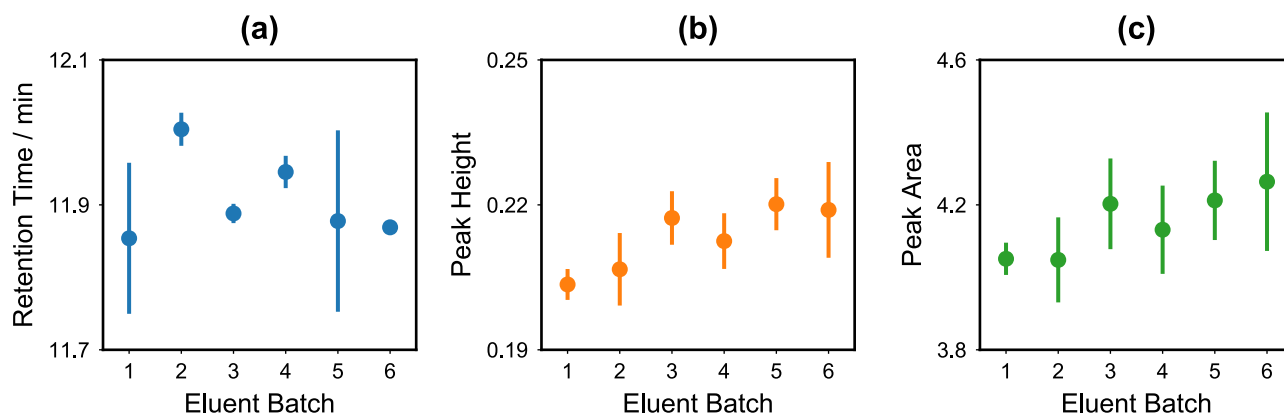


Figure 2.12: Plots showing the variation in (a) retention time, (b) peak height, and (c) peak area for measurements of a 1 mg/ml C_{60} solution in *o*-DCB with different batches of eluent solvent.

The results of this test (Fig. 2.12) suggested that there was some variation in peak retention time, height, and area between batches of eluent solvent, but also for measurements using the same batch of eluent. Overall, the mean peak retention time was 11.9 min, with a standard deviation of 0.08 min, the mean peak height was 0.213, with a standard deviation of 0.009, and the mean peak area was 4.152, with a standard deviation of 0.143. While all of these values have some variability, the standard deviations

were all relatively low, suggesting that the self-built HPLC gave sufficiently repeatable measurements for the intended experiments.

In order to test whether changing the batch of eluent solvent did increase the variability in these measurements, and if the reproducibility of the HPLC system could be improved further by using one large batch of eluent, the between-batch and within-batch variances were partitioned using a one-way analysis of variance (ANOVA). Due to the requirement for high reproducibility for the HPLC measurements, a confidence level of 99.5 % was used to judge whether differences between the two variances were statistically significant.

In this experiment, the between-batch mean squared error (MSE) for peak times was 0.020 min, compared to 0.005 min for the within-batch MSE. The probability that this difference was due to chance (the p-value) was 0.005, making it equal to the threshold probability for the chosen confidence level and so statistically significant. However, both MSEs were very small. As such, even though changing the batch of the eluent solvent gave an increase in the variance of the retention time, the overall variance was not large enough to suggest a significant change in composition of the eluent solvent between batches.

For peak height, the between-batch MSE was 0.0002 and the within-batch MSE of 0.00004, giving a p-value of 0.0003. For peak area, the between batch MSE was 0.05 and the within batch MSE was 0.016, giving in a p-value of 0.024. Both of these p-values are statistically significant, confirming an increase in variance in the measurements caused by changing the batch of eluent solvent. However, like the retention time, the overall standard deviations in the peak height and peak area were low. Therefore, although using different batches of eluent solvent did increase the error in measurements made by this HPLC system, the overall variance in the measurements was low. Despite this, erring on the side of caution, the HPLC system was changed to use a very large (5 L) feedstock of eluent solvent to ensure as low variation between measurements as possible, and to prevent the need to change it in the middle of longer experiments.

HPLC System Calibration

The HPLC system was calibrated to allow the relative proportions of the different fullerene derivatives to be extracted from a chromatogram. To do this, calibration curves were made by measuring chromatograms of samples of X_0 , X_1 , X_2 , and X_3 in *o*-DCB at concentrations of 0, 0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 1, and 2 mg/mL, repeating each measurement six times. Standards were prepared for these calibrations from samples of X_1 , X_2 , and X_3 synthesised and purified in batch following the above procedures, and from commercially bought X_0 . Stock solutions were made by dissolving 20 mg of each of the solids in 10 mL of *o*-DCB to give 2 mg/mL solutions. These stock solutions were then diluted to make solutions at the other required concentrations.

As the signal in the chromatogram at a given retention time is proportional to the concentration of the species eluting at that time, both the peak height and the peak

area can be used to calculate the concentration of a species in a sample. The peak area is more commonly used, as changes in the conditions of a HPLC system can lead to changes in the width of a chromatogram peak, which would affect the peak height but not the peak area. However, results from the previous section suggested that there was low variation in measurements of both peak height and peak area, even when different batches of eluent solvent were used. As such, calibration curves were made for both measures in order to decide which gave more accurate results.

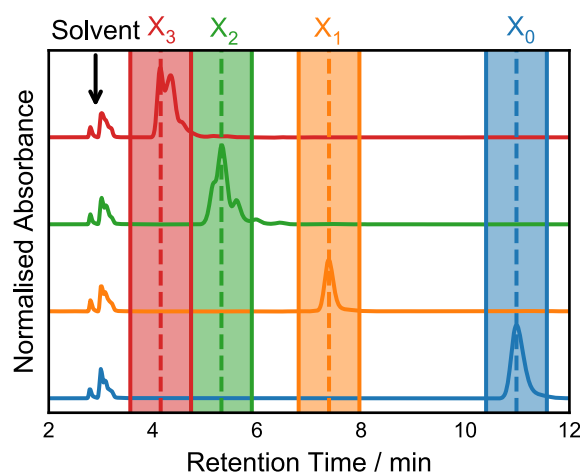


Figure 2.13: Examples of pure chromatograms for each adduct with the position of the peak maxima indicated by dashed lines, and the peak integration windows indicated by shaded areas.

As previous results showed that there was little variation in retention time of the chromatographic peaks, the peak height was calculated for each derivative by finding the maximum signal within a fixed window on the chromatogram. The peak areas were calculated by integrating the chromatogram signal over a fixed window about these peak maxima. In Figure 2.13, the position of each peak maximum is shown, as well as the fixed window about each maximum used for integrating each peak. Looking at the boundary between the X_3 and X_2 peaks, it can be seen that there is a small overlap of the X_3 peak into the integration window of the X_2 peak. This could cause calibrations based on the peak area to inaccurately estimate the concentration of X_2 and X_3 in product mixtures. As such, a calibration using the peak heights might be preferable if there is not too much variation in the peak shapes.

There is a linear relationship between chromatographic peak area and the concentration of a species. Assuming the peak height increases linearly with the peak area (i.e. the shape does not change with concentration), there is also a linear relationship between peak height and concentration. Therefore, ordinary least-squares regression can be used to calculate the relationship between peak height or area and concentration. However, ordinary least-squares is only valid if the variance of the measured peak height or area is

independent of the concentration being measured. If the variance changes with concentration, a weighted least-squares is necessary.

In order to determine if a weighted regression was necessary, the standard deviations of the measured peak heights (Figure 2.14a) and areas (Figure 2.14b) were plotted against concentration. From visual inspection of these plots, the standard deviation of both peak height and area definitely increased with increasing adduct concentration for X_1 and X_3 . However, the plots for X_0 and X_2 are not as clear, and the peak heights and peak areas may have had a constant standard deviation.

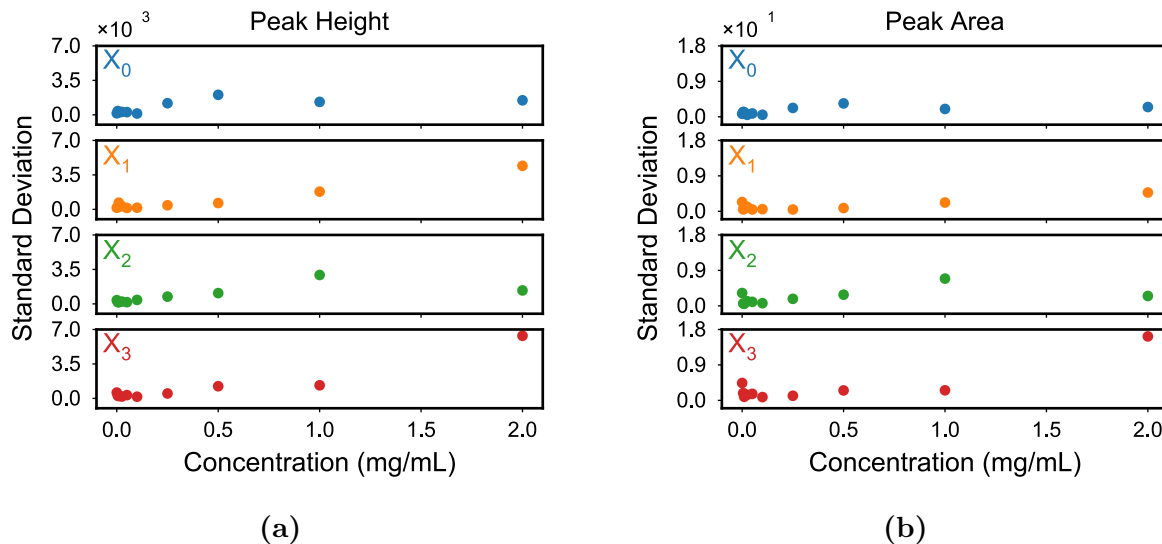


Figure 2.14: Standard deviations at each concentration used in for the calibrations by (a) peak height and (b) peak width

An F-test was carried out to determine if there was a statistically significant change in the measured peak height and area with the concentration of any of the adducts, involving comparing the ratio of the largest and smallest standard deviations from each set of measurements to a random distribution. For both peak height and peak area, all of the tests resulted in p-values that were much smaller than 0.001, implying that for all adducts the standard deviation of measured peak height and peak area changed with adduct concentration. Therefore, weighted least-squares regression was used to calculate the HPLC calibration curves.

Weights proportional to the concentration were used, and were scaled to sum to unity for each calibration curve. Their calculation is shown in Equation 2.3, where w_i and s_i are the calculated weight and variance of measurement i , and n is the total number of measurements.

$$w_i = \frac{s_i^{-2}}{\sum_i s_i^{-2}/n} \quad (2.3)$$

The resultant calibration curves are shown in Figs. 2.15a and 2.15b for the peak heights

and peak areas respectively. The peak height calibrations had very low intercept values, of 0.0005, 0.0005, 0.0003, and 0.0005 for X_0 , X_1 , X_2 , and X_3 , and gradients of 0.1265, 0.0835, 0.1380, and 0.1213 respectively. Both the gradients and intercepts were higher for the peak area calibrations, at 0.0065, 0.0047, 0.0101, and 0.0054 for the intercepts and 2.2658, 1.0720, 2.7646, and 2.6462 for the gradients of the regression lines for X_0 , X_1 , X_2 , and X_3 , respectively. Furthermore, when there is no adduct present, as in the blank measurements, taking the integral of the baseline will return a larger intercept than taking the height of the baseline at a single retention time.

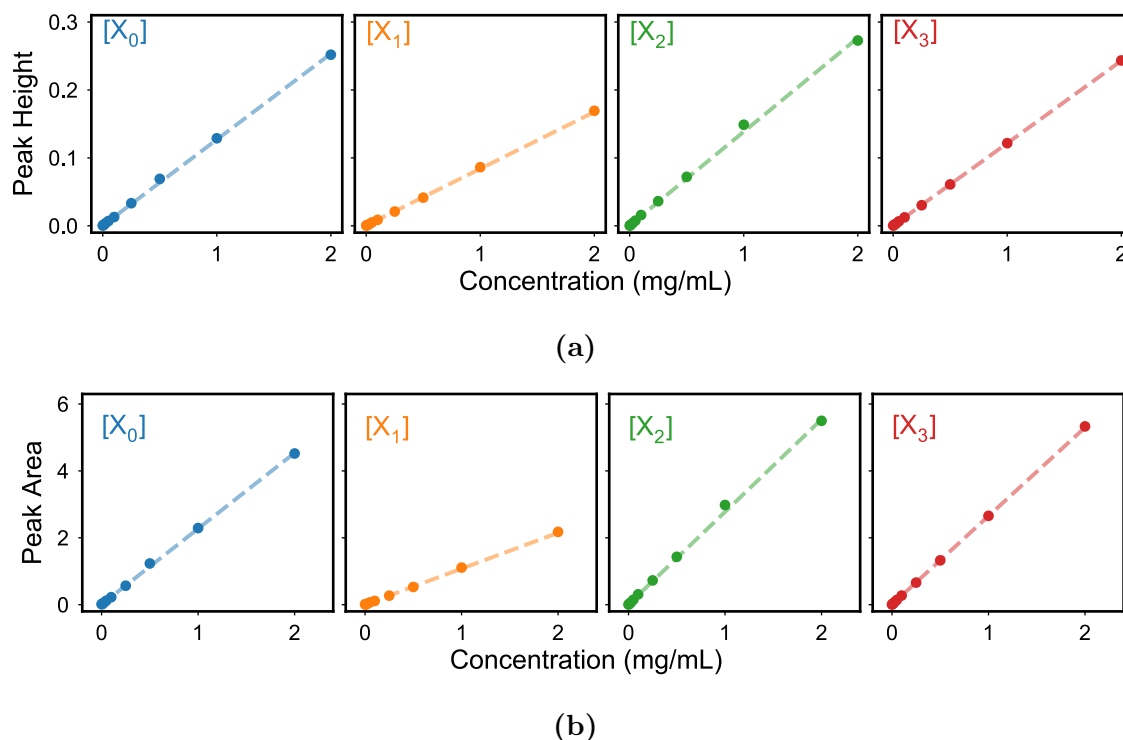


Figure 2.15: Calibration curves for relating (a) peak height and (b) peak area to adduct concentration.

A more informative comparison is to calculate the limits of detection for each calibration method. The limit of detection is usually taken as the average value of the blank measurement (y_B), plus its standard deviation (s_B) multiplied by some constant (k), as shown in Eqn. 2.4.

$$\text{Limit of Detection} = y_B + ks_B \quad (2.4)$$

The limit of detection is usually defined as the lowest concentration at which a species will show in a chromatogram, and for this the constant k is often set to 3. The lower limit at which the amount of species can be reliably determined, the limit of quantification, will be higher than this, and so a value of 10 is used for k . These limits were calculated for both calibration methods, and are shown in Table 2.2.

	Limit of Detection / (mg/mL)		Limit of Quantification / (mg/mL)	
	Peak Height	Peak Area	Peak Height	Peak Area
X ₀	0.0038	0.0102	0.0125	0.0339
X ₁	0.0058	0.0662	0.0192	0.2205
X ₂	0.0082	0.0354	0.0272	0.1179
X ₃	0.0146	0.0500	0.0487	0.1667

Table 2.2: Calculated limits of detection and quantification for calibrations by peak height and area

This shows that the calibration using peak area gave higher (worse) limits of detection and quantification for all fullerene derivatives. In the best case, for X₀, they were only around 3 times higher than the limits for the peak height calibration: 0.0102 and 0.0339 mg/mL for the limit of detection and quantification respectively, compared to 0.0038 and 0.0125 mg/mL when using peak heights. However, for X₁ the limits were around 10 times higher than for the peak height calibration: 0.0662 and 0.2205 mg/mL for the peak area calibration but only 0.0058 and 0.0192 mg/mL for peak height. Clearly it is better to have as low a limit of quantification as possible, and therefore the peak height calibration was the better method for calculating fullerene derivative concentrations, giving a limit of quantification below 0.05 mg/mL for all adducts.

Another important factor for deciding between the two methods for calculating concentration was the prediction errors that they gave. Calibration curves like these must be used inversely to predict the concentration of an adduct - a measurement of the y-value must be used to calculate the x-value. As such, the calculation of the concentration prediction error proceeds by first calculating the error in the y-value from the calibration, $s_{x/y}$, shown in Equation 2.5, where y_i is the measured peak height or peak area of measurement i in the calibration, $\hat{y}_i$ is the corresponding value from the calibration fit, and n is the total number of measurements used for the calibration.

$$s_{x/y} = \sqrt{\frac{\sum_i (y_i - \hat{y}_i)^2}{n - 2}} \quad (2.5)$$

$$s_{x0} = \frac{s_{x/y}}{b} \sqrt{\frac{1}{w_0} + \frac{1}{n} + \frac{(y_0 - \bar{y})^2}{b^2 \sum_i (w_i x_i - \bar{x})^2}} \quad (2.6)$$

The prediction error of the x-value, s_{x0} , can then be calculated by the formula shown in Equation 2.6. In this equation y_0 is the measured peak height or peak area for the sample, w_0 is the corresponding weight for that value found by interpolating between the weights

used in the calibration, b is the slope of the calibration, n is the number of measurements used in the calibration, $\bar{y}$ is the mean fitted value in the calibration being used, $\bar{x}$ is the mean concentration used in the calibration, and x_i is the concentration of measurement i for the calibration, with w_i as the corresponding weight of that measurement. Prediction errors were calculated using this formula for both calibration curves, and used to calculate 99.9 % confidence bounds. The calibration curves with these confidence bounds are shown in Fig. 2.16a for peak height, and Fig. 2.16b for peak area.

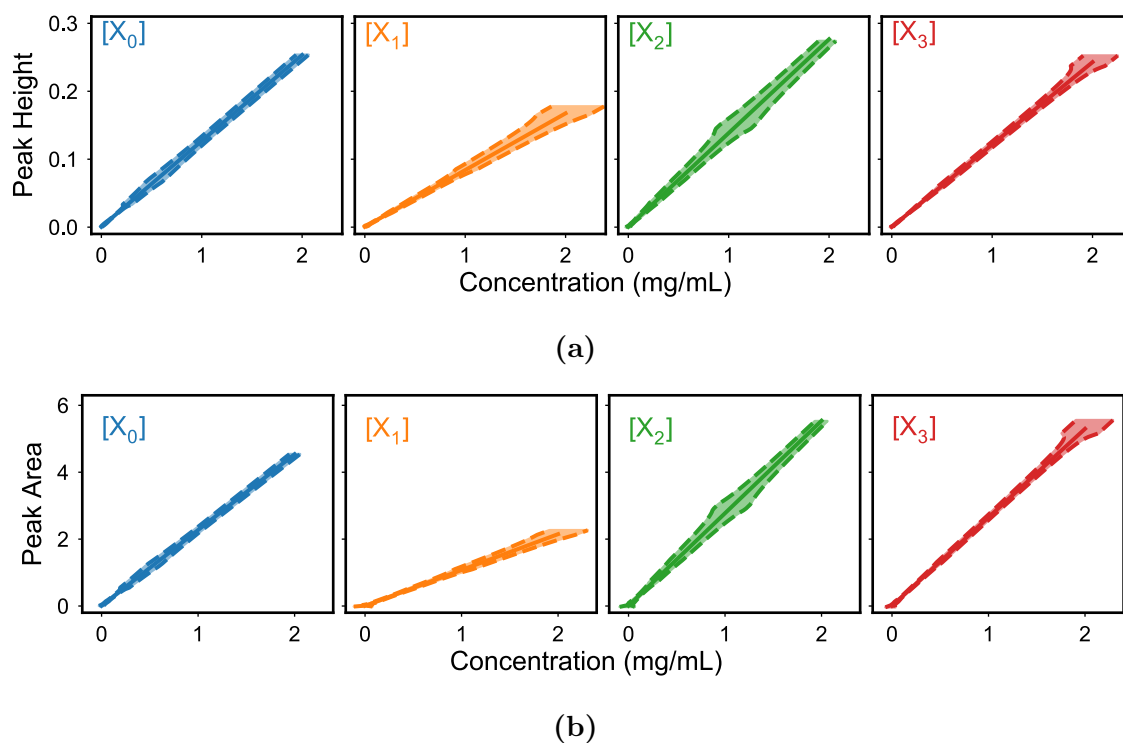


Figure 2.16: Calibration curves for relating (a) peak height and (b) peak area to the adduct concentration, showing the 99% confidence bounds as a shaded area about the regression line.

The confidence bounds look similar for both calibration methods, showing increasing uncertainty in the concentration predictions as the measured signal increases. Therefore, both methods for calculating the concentration of the fullerene derivatives appear to have similar prediction errors. This is evident when comparing the prediction error in the concentration of each adduct for the peak height and peak area values in the middle of the ranges measured during the calibration. For peak height, these prediction errors were 0.013, 0.029, 0.043, and 0.01 mg/mL for X_0 to X_3 respectively, while the corresponding prediction errors for peak area were 0.016, 0.033, 0.042, and 0.01 mg/mL. Given the lower limits of detection for the peak height calibration, and that it was slightly simpler to calculate, the peak height was chosen as the method for calculating the proportions of *o*-xylenyl functionalised fullerene derivatives from the chromatograms.

HPLC Calibration Test

The calibration for calculating the fullerene derivative concentration using chromatogram peak height was tested by using it to estimate the concentration of fullerene derivatives in ten samples of known composition. The results of this test are shown in Fig. 2.17(a - d), with the known concentration of the fullerene derivatives in the samples shown in black, and the measured concentrations shown in the corresponding colour with 99.9 % confidence error bars.

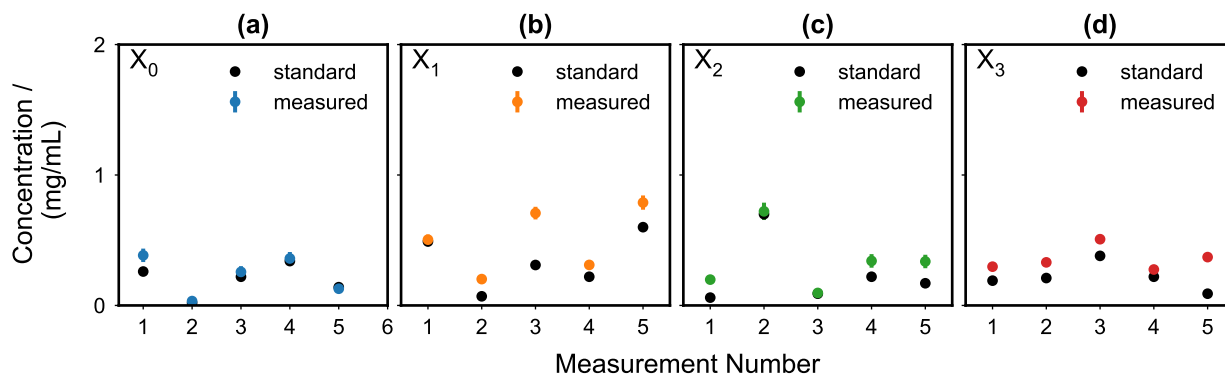


Figure 2.17: Comparison of the concentration of (a) X₀, (b) X₁, (c) X₂, and (d) X₃ in test samples, measured using the calibrated HPLC system to the concentration that the test samples were made up to.

All of the measured concentrations were close to the concentrations of the standard samples, especially for X₀ and X₂. However, the differences between the measured and standard concentrations were larger than the error in prediction calculated from the calibrations. The largest differences were seen for X₃, where 4 of the 5 measurements had differences larger than the prediction error, and for X₁ where there was a significant difference from the expected concentration for measurement 3.

This suggests that the concentration measurement error was higher than just the small prediction error calculated for the HPLC calibration. The mean squared error of the measured concentration in these samples was 0.012, 0.044, 0.004, and 0.025 mg/mL for X₀, X₁, X₂, and X₄ respectively. These larger errors were most likely due to changes between the eluent solvent used for the calibration and for these measurements, as well as errors in the making the samples up to the desired concentrations. However, using the MSE values from this test, the error in the measured concentrations was still acceptably low overall, making the HPLC system suitable for use in the automated reactor as in-line analysis.

2.3.3 Constructing the Complete Flow Reactor

The complete automated flow reactor was composed of a dual-channel continuous-flow syringe pump (Syrris Asia) for reagent delivery, a self-built solid state heater, and a self-

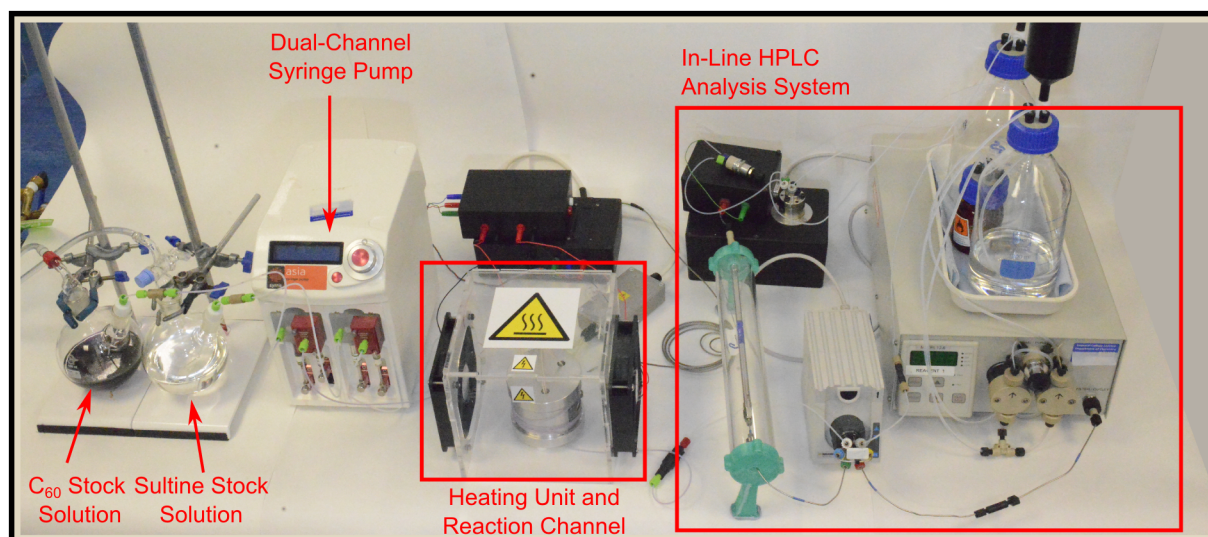


Figure 2.18: Photograph of the complete automated continuous flow reactor for the synthesis of *o*-xylenyl functionalised fullerene derivatives, with the reagent stock solutions, syringe pump, heating unit, reaction channel, and in-line HPLC analysis system labelled.

built in-line HPLC analysis system. The two reagent solutions, C₆₀ in *o*-DCB (2 mg/mL) and sultine in *o*-DCB (1.4 mg/mL), were stored in two-neck round-bottom flasks (500 mL) under argon. The second necks of the round-bottomed flasks were fitted with custom PTFE stoppers, designed and fabricated by Mr. Simon Turner, through which a piece of PTFE tubing could be passed into the solution and fastened in place with a microfluidic fitting. The two solutions were merged using a static y-shaped mixer (PEEK, Upchurch) before entering the reaction channel, wound helically around the solid state heater. A back-pressure regulator was placed at the outlet of the reaction channel, and in turn connected to the HPLC analysis. The total length of tubing in the reactor was 202 cm, for a total reaction volume of 1.587 mL, while the length of tubing in contact with the heater was 119 cm, for a heated reactor volume of 0.931 mL.

2.4 Automated Flow Synthesis of Fullerene Adducts

2.4.1 Stock Solution Preparation

Stock solutions of C₆₀ (2 mg/mL) and sultine (1.4 mg/mL) were prepared for the flow reactor by dissolving the as-received C₆₀ and the un-purified sultine, synthesised according to the procedure above, in *o*-DCB under argon. When the reactor was not being operated, both solutions were stored under argon. Degradation of the sultine stock solution occurred slowly over time at room temperature, by the formation and then polymerisation of *o*-QDM, and so was stored in the freezer when not used for a synthesis.

2.4.2 Automated Operation

The heater, HPLC system, and syringe pump were controlled by a computer, across the Universal Serial Bus (USB). For automated operation, a MATLAB function was written that accepted a list of temperatures and flow rates as the input. The script would then set the reactor running at each set of conditions and analyse the composition of the product formed using the in-line HPLC system.

The script stepped through each set of conditions in turn. At each step, the set-point of the heater was updated first. Once it was at the desired temperature, the two syringe pumps were set to the specified volumetric flow rates. The flow was allowed to stabilise for a duration equal to twice the mean residence time, given by Eqn. 2.7:

$$\Delta t = 2 \frac{V}{Q_{C60} + Q_{Sultine}} \quad (2.7)$$

where Δt is the stabilisation time, V is the volume of the reactor, and Q_{C60} and $Q_{Sultine}$ are the flow rates set on the C₆₀ and sultine syringe pumps. During this time, the injection valve of the HPLC system was kept in the ‘inject’ position, with the sample loop between the eluent stream and the column, so that the product flowed directly into a collection flask. After waiting for the flow to stabilise, the injection valve was then switched to the ‘load’ position, with the sample loop between the product stream and collection flask, for a length of time sufficient for 10 volumes of the sample coil (50 μ L) of fluid to pass through. With the product loaded in the sample coil, the valve was switched back to inject it into the HPLC system for analysis.

To efficiently use time and reagents, the syringe pumps were stopped when the product was injected into the HPLC system, and the next set of conditions started while it was being analysed. This was done by, again, setting the temperature and waiting for it to stabilise. The syringe pumps were started to infuse at the new flow rates when the remaining time left for the analysis of the previous product was less than or equal to the Δt needed for the new flow rates.

All of the signals acquired for each product were saved to a file by the function. After running all of the supplied conditions, the signals were converted to chromatograms and the mole fractions of the fullerene derivatives were calculated and returned to the user. A file containing the conditions explored and the measured mole fractions was also saved. The code for this function is provided in the Appendix.

2.4.3 Automatic Optimisation

During automated optimisations of the synthesis of *o*-xylenyl functionalised fullerene derivatives, the stock solutions were made as above. The optimisations were carried out using a custom-written class-based wrapper around Huyer and Neumaier’s⁹⁵ original

MATLAB implementation of SNOBFit. The wrapper was written as a MATLAB package, which allowed an easy interface for running both the unconstrained and constrained SNOBFit optimisations, described in later chapters. The objective and constraint functions were written as MATLAB function files, that the SNOBFit object could access. These functions used the previously described MATLAB function to operate the flow reactor.

The code for the SNOBFit class package is provided at: https://github.com/jdmgroup/SNOBFit_for_chemical_optimisations. Improvements to the file handling of the code hosted on Github were made by Dr. James Bannock. There is also a set of tutorials on how to use the code to run constrained and unconstrained chemical optimisations, which were developed with the help of Dr. James Bannock and Prof. John de Mello. An example of the objective and constraint function definitions are provided in the Appendix.

2.5 Flow Synthesis of Iron Oxide Nanoparticles

The final chapter of this thesis describes the construction and development of a flow reactor for the synthesis of magnetic iron oxide nanoparticles. This was preliminary work towards the aim of controlling the size, and thereby magnetic properties, of the nanoparticles through tuning the reaction conditions in flow. A full description of the development of the flow reactor is provided in Chapter 5, while a description of the batch syntheses of iron oxide nanoparticles and the reagents used in the flow syntheses are provided here.

The thermal decomposition of $\text{Fe}(\text{acac})_3$ was chosen as the synthetic route because previous reports in the literature suggest that it produces higher quality nanoparticles with a greater degree of control over particle size than other routes. The synthesis was originally tried in flow using the route described by Zhen Li *et al*¹⁰², with 2-pyrrolidone as both reaction solvent and capping ligand. After this proved unsuccessful, an alternative route was used with diphenyl ether as the reaction solvent¹⁰³. Both of these routes were tested in batch to confirm that they formed magnetic iron oxide nanoparticles, before attempting to adapt them to flow.

2.5.1 Batch Synthesis of Iron Oxide Nanoparticles

Synthesis of Iron Oxide Nanoparticles in 2-Pyrrolidone

The batch synthesis of iron oxide nanoparticles via the route described by Zhen Li *et al*¹⁰² was carried out by dissolving $\text{Fe}(\text{acac})_3$ (706 mg, 2 mmols) in 2-pyrrolidone (20 mL) under argon. Once fully dissolved, the reaction mixture was heated to reflux (245 °C) and stirred for ten minutes. During this time, the solution changed from a bright red colour to black, indicating the formation of iron oxide nanoparticles. The reaction mixture was

then allowed to cool to room temperature. Upon addition of methanol, a dark brown precipitate formed. This was cleaned by centrifugation and washing with acetone three times to give a dark brown solid. Upon re-dispersal in water, the nanoparticles displayed (super)paramagnetism in the form of attraction to a strong magnet.

Synthesis of Iron Oxide Nanoparticles in Diphenyl Ether

While the previous synthetic procedure was simple to perform in batch, in flow it caused unavoidable fouling of the reaction channel, hypothesised to be a result of 2-pyrrolidone preferentially wetting the reaction channel walls at high temperatures. An alternative synthesis was therefore developed using an alternative reaction solvent. ODE has previously been used successfully in high temperature nanoparticle syntheses²², and demonstrated good flow characteristics. However, $\text{Fe}(\text{acac})_3$ is not soluble in ODE, and so a reaction solvent that was a mixture of ODE and another solvent that dissolved $\text{Fe}(\text{acac})_3$ was needed. A synthetic procedure that used diphenyl ether as the reaction solvent was chosen, due to the controllable nature of the synthesis and the miscibility of ODE and diphenyl ether.

This procedure was tested in batch by dissolving $\text{Fe}(\text{acac})_3$ (706 mg, 2 mmol) in diphenyl ether (20 mL), and adding oleyl amine (1.3 mL, 6 mmol), oleic acid (1.51 mL, 6 mmol), and 1,2-hexanediol (2.58 g, 10 mmol). The reaction mixture was then heated to reflux (260 °C) under argon and stirred for 30 min. Over this time the solution changed from a bright red colour to a dark brown. It was then cooled to room temperature, and ethanol was added to precipitate the nanoparticles. This product was redissolved in hexane and then precipitated with ethanol. The resulting brown solid exhibited attraction to a strong magnet, indicating the nanoparticles were paramagnetic.

2.5.2 Synthesis of Iron Oxide Nanoparticles in a Three-Phase Flow Reactor

Reagents for the Flow Synthesis of Iron Oxide Nanoparticles in 2-Pyrrolidone

The concentrations of reagents in the flow synthesis was half of that used in the batch synthesis, as a precaution against reactor fouling. The reagent mixture was made by dissolving $\text{Fe}(\text{acac})_3$ (353 mg, 1 mmol) in 2-pyrrolidone (20 mL). Once no more solid could be seen, the mixture was loaded into a glass syringe and the reactor operated as described above.

Reagents for the Flow Synthesis of Iron Oxide Nanoparticles in Diphenyl Ether

The reagent mixture was made using a similar procedure as in the batch synthesis, but with the solvent changed to a mixture of ODE and diphenyl ether to give good droplet

flow characteristics in the reactor, and the reagent concentrations halved. $\text{Fe}(\text{acac})_3$ (353 mg, 1 mmol) was dissolved in a 1:1 mixture of ODE (10 mL) and diphenyl ether (10 mL), with oleyl amine (0.65 mL, 3 mmol) and oleic acid (0.755 mL, 3 mmol). The alcohol, 1,2-hexanediol, was omitted because in a first trial it formed a solid deposit in the syringe. When fully dissolved, the reaction mixture was loaded into a glass syringe and the reactor operated as described above.

2.5.3 Droplet Analysis

No in-line analysis was developed for this flow synthesis. However, while developing the flow reaction, the success of the synthesis was measured using video analysis of the droplets. All videos were taken using the ambient lighting of the laboratory, and at 25 fps. The analysis was carried out using the Python programming language, and is described in Chapter 5.

CHAPTER 3

**Automation and Optimisation of the
Cascadic Synthesis of Fullerene
Derivatives**

3.1 Introduction

Cascadic reactions provide a challenge for optimisation. A series of consecutive competitive reactions consume each product as they are generated, to form the next product in the cascade. Depending on the desired product, and the kinetics of the reaction, the resultant parameter space may be complex, and lead to an unavoidable distribution of products. In some cases it is not only necessary to maximise the yield of a desired product but it is also important to suppress the formation of certain other species that may be detrimental to the intended application.

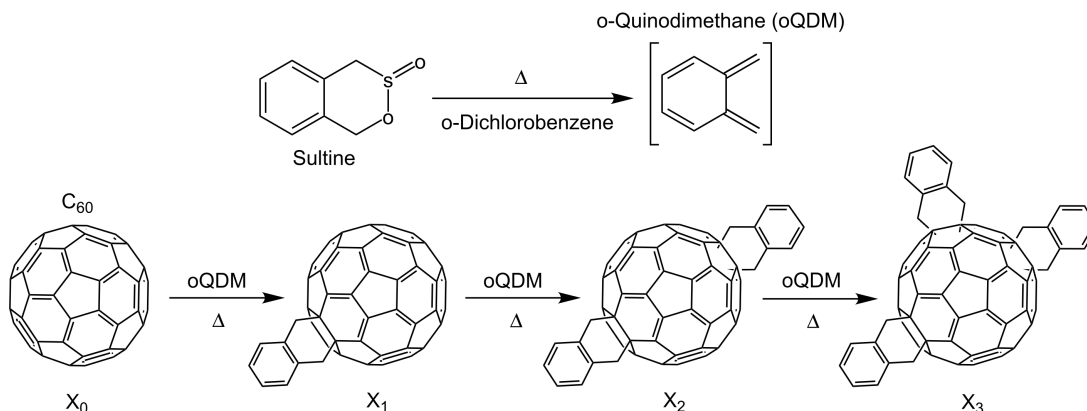


Figure 3.1: Synthesis of *o*-xylenyl functionalised C_{60} derivatives from the Diels-Alder cycloaddition of *in-situ* formed *o*-quinodimethane to C_{60} fullerene.

The synthesis of *o*-xylenyl C_{60} adducts⁹⁹ is an example of a functional material that is produced by a cascadic reaction. In this synthesis, sultine is heated with C_{60} in *o*-dichlorobenzene. This forms the reactive intermediate *o*-quinodimethane (*o*QDM) *in-situ*, which subsequently attaches to the C_{60} once to form the singly-functionalised product, X_1 , by Diels-Alder cycloaddition. X_1 is then consumed by further reaction with the *o*QDM, forming the doubly-substituted product, X_2 , which in turn is consumed to give the triply-substituted product, X_3 , and so on. The reaction scheme is shown in Fig. 3.1

Due to the cascadic nature of the reaction, combined with the similar reactivity of each of the functionalised fullerenes, the product is a mixture of adducts. Both the singly and doubly functionalised species, X_1 and X_2 , can be used as electron acceptors in the active layer of bulk heterojunction solar cells, with X_2 giving the best device performance^{99,104,105}. However, the triply substituted product gives very poor device performance, and even small amounts left in the product can dramatically reduce solar cell efficiencies¹⁰⁶. Therefore, the best outcome for this reaction would be to optimise to produce a high yield of X_2 , with minimal X_3 .

However, given how similar the products are, finding the right combination of reaction conditions would require extensive experimentation. Done using traditional “batch” chemistry, this would be very time and labour intensive. Not only would a large number

of experiments need to be run, but there is no guarantee that the optimum set of conditions would be found. In this case, developing an automated flow reactor is an attractive solution. As well as the increased control and reproducibility offered by flow methods, using an automated reactor allows a large number of experiments to be run without the need for human intervention. Furthermore, incorporating in-line analysis opens up the possibility of running intelligent optimisations, to more efficiently optimise the synthesis than could be done manually.

A number of automated flow reactors have been developed over the past decade^{78–84,87–89}, targeting a number of different products. However, there are very few examples of automated flow reactors used for the synthesis of functional materials, with one notable example used to tune the emission wavelength of chalcogenide quantum dots²¹. Similarly, a continuous-flow synthesis of related indene functionalised fullerene derivatives has been reported¹⁷, but this work focused on adapting the synthesis to a flow system rather than optimising the reaction. Given the need for a high degree of control over reaction conditions, and the tuneable properties of many materials, there is a significant gap in the applications of automated flow reactors to the different challenges of functional material synthesis. The focus of this work was to develop an automated flow reactor to optimise a cascadic synthesis of fullerene derivatives, addressing a significant challenge that has not been focused on before.

3.2 Flow Reactor Design and Automation

There were a number of design choices in building the flow system, that were informed by the need to incorporate in-line analysis and automate the reactor, as well as the need to provide a stable enough platform to run experiments over many days. One such design choice was whether to use a commercial or self-built flow reactor. Although commercial systems can often be used with minimal set-up, and have been used previously for the flow synthesis of functionalised fullerenes¹⁷, they tend to be limited in what they can be used for. This is especially true for automated systems, where the user will often be locked into using the manufacturer’s proprietary software, making deviations from the intended use difficult. Indeed, the work described in this chapter builds on a flow synthesis of *o*-xylenyl functionalised fullerene derivatives that was developed during my MRes project using a commercial “FlowSyn” system, where automation and therefore self-optimisation was made virtually impossible by the limitations of the equipment.

As a reaction to the inflexible and expensive nature of commercial flow equipment, there has been significant work within many research groups to develop simple, customisable flow systems that can be built from readily available parts⁷. Within my own research group, flow reactors have been built for the synthesis of optoelectronic polymers with tailored molecular weights¹⁹, the high-temperature synthesis of nanocrystals²², the

controlled multi-step synthesis of large quantum dots⁵⁷, and the large-scale synthesis of metal chalcogenide quantum dots²⁴. Encouraged by this work, I elected to build my own automated flow system. The overall design of the system incorporated syringe pumps to deliver the reagents to a PTFE tubing reaction channel, a unit for heating the reaction channel, and an in-line analysis system. A description of the development and construction of the automated flow reactor can be found in Chapter 2.

3.2.1 Choice of In-line Analysis

A number of different analytical techniques have been adapted for in-line use with automated flow reactors, including infrared absorption⁸⁹ and visible fluorescence spectroscopy²¹, nuclear magnetic resonance⁷², mass spectrometry⁸⁷, and liquid^{78,80–82} and gas chromatography^{79,83,84}. Of these techniques, liquid chromatography was chosen as the most suitable for measuring the composition of *o*-xylenyl fullerene adduct mixtures. While visible absorption spectroscopy and mass spectrometry have previously been used to characterise these fullerene adducts, the similarity in absorbance spectra between the adducts and the cost of adapting a mass spectrometer to a flow system precluded their use in this case.

HPLC, on the other hand, has been shown to give good resolution between the adducts produced in this synthesis^{99,105}. In an HPLC system, a sample is injected into a flow of eluent solvent that carries the sample through a column packed with functionalised silica. The different species in the sample are separated by their relative affinity to the column: species with a greater affinity to the column than the eluent solvent take a longer time to pass through. The absorbance of the outflow from the column is measured, often at a single wavelength, to detect the peaks as they elute. The resultant signal creates a chromatogram, with peaks at different times representing different species, and the area under each peak being proportional to the molar absorptivity and concentration of that species in the sample. In this way, HPLC analysis of the flow reactor product mixture can give information about the composition of the product mixture.

The in-line HPLC system was constructed as described in Chapter 2. An eluent solvent was pumped through a functionalised silica column, where a purpose built detector measured the transmitted light at a single wavelength at the exit of the column. A switching valve was used to inject the sample into the system and to trigger the detector to start measuring. The eluent used was a 3:1 mixture of toluene/hexane and the (commercially sourced) column was packed with pyrenylpropyl-functionalised silica, both chosen from analyses of *o*-xylenyl derivatised fullerenes in the literature^{99,101,105}.

3.3 Automated Flow Synthesis

The construction of the complete flow reactor developed for the automated synthesis of *o*-xylene functionalised fullerene derivatives is described in Chapter 2, and the system is shown as a schematic in Figure 3.2. Moving from left to right, syringe pumps were used to deliver the reagents to the reaction channel, through tubes that met at a y-shaped mixer. The reaction channel itself was made PTFE tubing with an internal diameter of 1 mm, wound around the self-built solid-state heater. The out-flow of the reaction channel was linked to the switching valve of the HPLC analysis unit, allowing samples of the reactor product to be injected for in-line analysis.

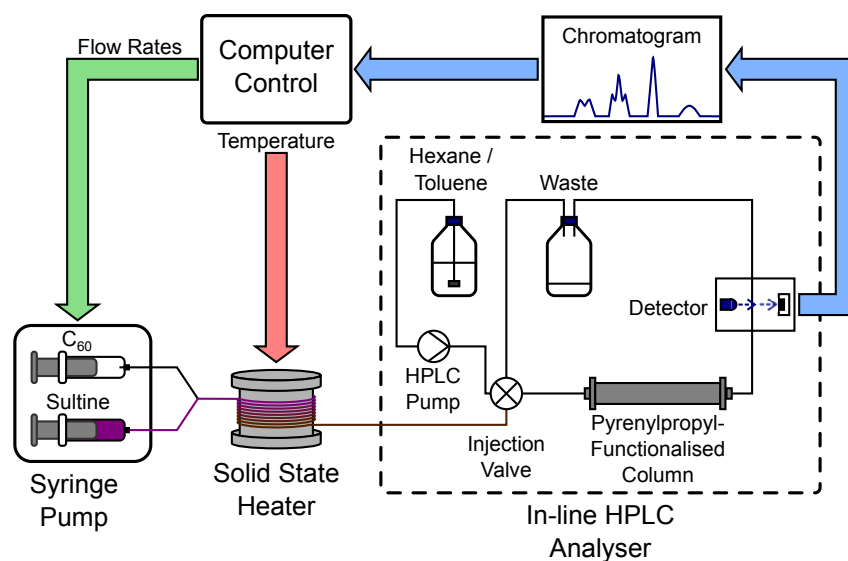


Figure 3.2: Diagram of the flow-reactor built for the synthesis of *o*-xylene C₆₀ adducts, showing the in-line HPLC system and computer control.

A continuous flow reactor was used as all reagents and products were soluble, meaning there was no risk of fouling. This had the benefit of avoiding the increased complexity of segmented flow reactors.

As shown in Figure 3.2, every part of the flow reactor was controlled by a central computer. This allowed all of the reaction parameters to be varied using the computer: the temperature could be varied by changing the set-point of the solid state heater, the average residence time by changing the overall flow rate of both syringe pumps, and the reagent ratio by changing the ratio of flow rates between the two syringe pumps.

The reactor was controlled by a function written in the MATLAB programming language, to make it easier to use and give reproducible behaviour. This reactor control function was written to use time and material efficiently, by running one point while the previous point was being analysed. This is described in more detail in Chapter 2, and the code used is provided in the Appendix.

Automating the flow reactor in this way opened up the possibility of running experi-

ments with many data points, or over a long period of time. Storing the reagent solutions under argon in large round-bottomed flasks, and having the syringe pumps draw directly from these into the reactor, also increased the number of the experiments that could be run. This allowed long and complex experiments to be carried out with minimal intervention. A typical experiment carried out by this system comprised between 100 and 200 measurements.

3.3.1 Reactor Stability and Reproducibility

Experiments were run to test the stability and reproducibility of the synthesis of *o*-xylenyl functionalised C₆₀ derivatives. If the result gained from a particular set of input parameters is different each time the function is evaluated, then the underlying function is changing and therefore can not be successfully optimised. Similarly, for results to be comparable between optimisations of the same synthesis, they need to be reproducible.

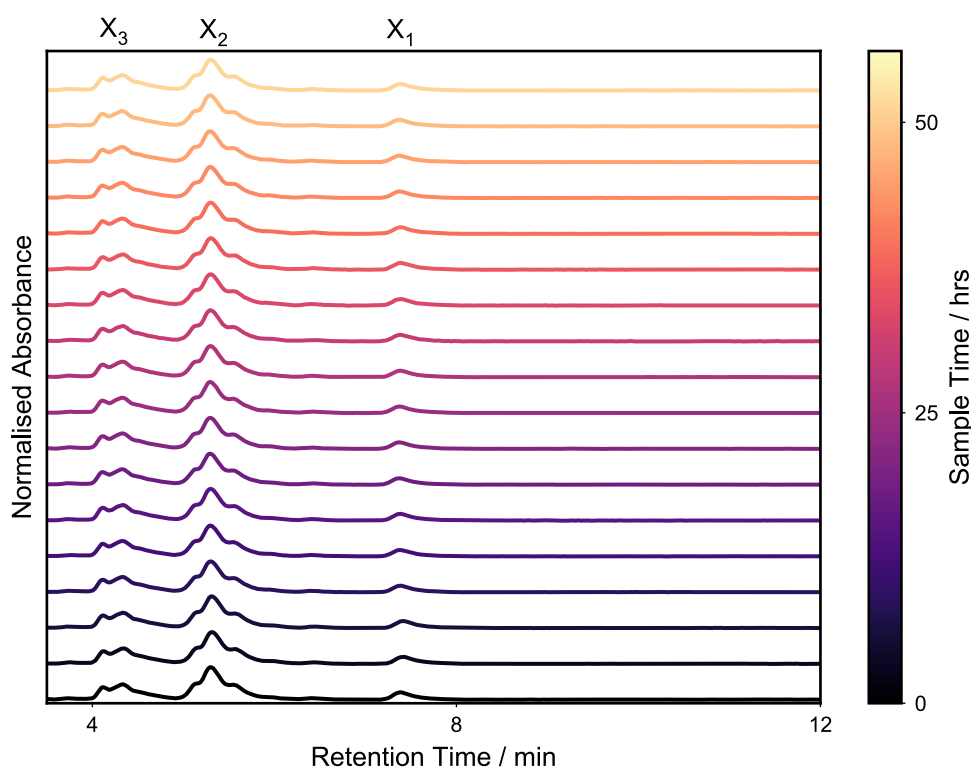


Figure 3.3: Chromatograms of the reactor product taken over the course of 50 hours at the same reaction conditions, showing good stability of the reactor.

To test the stability of the reactor, different product mixtures synthesised using the same set of reaction conditions were analysed. An experiment was carried out where a product mixture was synthesised every three hours at a temperature of 120 °C, a residence time of 30.7 min, and a [Sultine]:[C₆₀] ratio of 5.2:1. The syringe pumps and heater of the

reactor were stopped between each synthesis, so the reactor started cold every time. The experiment was run for 51 hours (just over two days) in which time 19 different product mixtures were made.

Chromatograms for all of the product mixtures made during this experiment are shown in Figure 3.3. The first chromatogram measured is at the bottom of the figure, with time increasing up to the final chromatogram, measured at 51 hours, at the top. Just by visual inspection, it can be seen that the product composition remained very stable over the course of the entire experiment. There appears to have been little variation in either the retention time or height of the peaks. The X_3 peak appears first, at 4.3 min, followed by the X_2 peak at 5.3 min, and finally the X_1 peak at 7.4 min. There is no X_0 peak, meaning that full conversion was achieved at these reaction conditions.

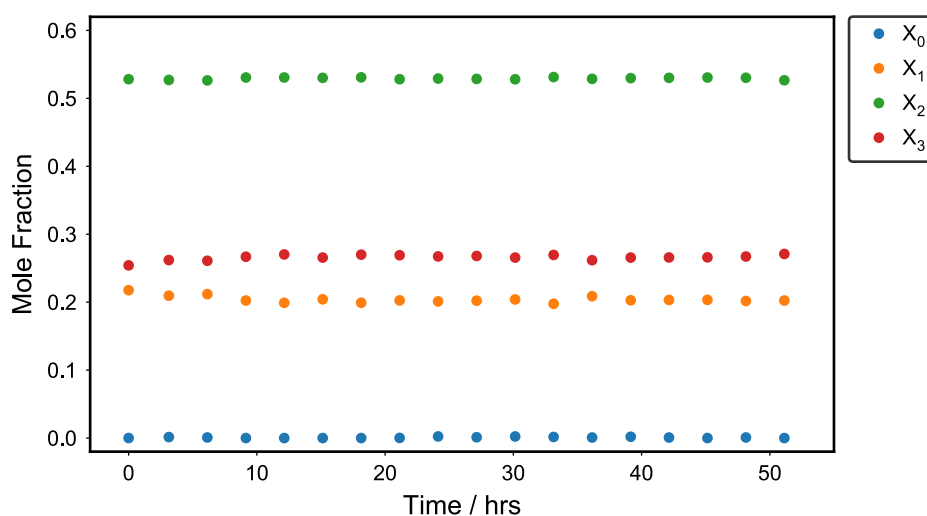


Figure 3.4: Mole fractions of the reactor product taken over the course of 51 hours at the same reaction conditions, showing the stability of the reactor.

A clearer picture of the stability of the product mixture can be gained by looking at the mole fractions of each of the adducts, as in Figure 3.4. The major product at these reaction conditions was X_2 with a mean mole fraction of 0.529, then X_3 with a mole fraction of 0.265, and finally X_1 with a fraction of 0.204. As previously noted, there was no unreacted X_0 left in the product.

The standard deviations for these mole fractions were 0.0041, 0.0015, and 0.0049 for X_3 , X_2 , and X_1 respectively. All of them were less than 1 %, demonstrating production of a consistent product at the same set of reaction conditions for over two days - long enough to run an experiment consisting of over 100 data points. As such, we can be confident that the underlying function relating the reaction conditions to the product composition will not change significantly over the course of an optimisation experiment.

Although the reactor demonstrated good stability during this test, a single set of reagent stock solutions was used for the whole experiment, leaving the possibility that a

change in stock solution might change the behaviour of the system. For instance, inaccuracies in the concentration of these stock solutions could alter the product composition obtained under nominally identical reaction conditions. The stock solution of sultine was most likely to have the greatest variation in concentration as a new batch was synthesised for each experiment, unlike the C_{60} , which was bought as a single batch from a commercial supplier.

The sultine was not purified before using it in the flow reactor, due to its low decomposition temperature coupled with the fact that it was an oil. As such impurities and some residual reaction solvent were left in the sultine when it was made into a stock solution for the flow reactor. To attempt to control for this, the purity of the sultine was estimated by NMR before making the stock solutions. However, this was not an accurate process and may have caused significant inaccuracies in the concentration of the sultine stock solution.

An experiment was run to determine how important batch to batch variation in the sultine stock solution might be. Five different batches of sultine were synthesised and used in the flow reactor to make product mixtures under the same set of reaction conditions. These batches of sultine were all synthesised and worked-up in the same way, their purity estimated by NMR, and the appropriate amount of *o*-DCB was then added to make them into 1.4 mg/mL stock solutions. The purities of the samples were determined to be 87.5, 91.5, 86.4, 82.8, and 74.5 %, giving an average purity of 84.5 % with a standard deviation of 5.7 %.

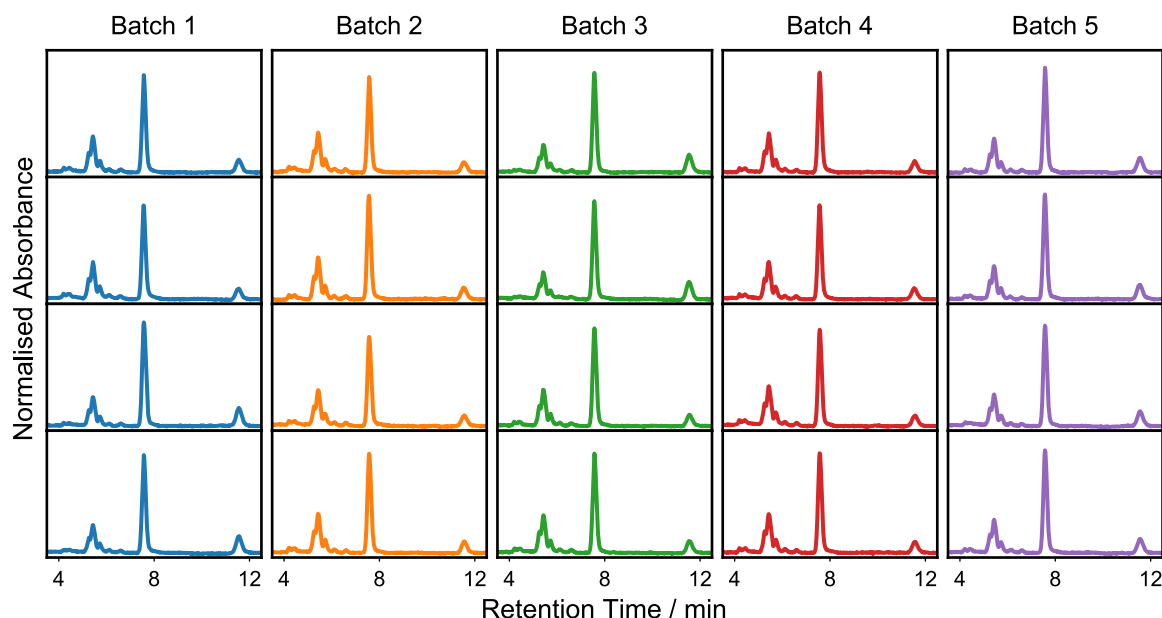


Figure 3.5: Chromatograms of the reactor product synthesised under the same conditions (temperature = 111 °C, residence time = 6.2 min, [Sultine]:[C_{60}] = 5.2:1), using five different batches of sultine and four repetitions of each batch.

The stock solutions were stored under argon and connected to a single syringe pump through different tubes using a ten-port selection valve, allowing a single sultine stock solution to feed the reactor at one time. A fifth port was connected to a flask of *o*-DCB, to clean the syringe pump and reaction channel between batches of sultine solution. The same stock solution of C₆₀ in *o*-DCB was used for the whole experiment.

The experiment was carried out in a randomised order and each sultine batch was tested four times. All samples were synthesised under the same reaction conditions, chosen randomly at the start of the experiment. The chosen temperature was 111 °C, the residence time was 6.2 min, and the [Sultine]:[C₆₀] was 5.2:1.

Each column in Fig. 3.5 shows the chromatograms produced by a single batch of sultine. For all of the measurements, the relative peak heights remain qualitatively the same - X₁ is the predominant product, with the highest peak, then X₂, followed by X₀, and finally X₃ with the smallest peak height. Within each batch it appears that there is almost no variation in the peak heights for each fullerene derivative. There also appears to be little variation in peak heights between the batches of sultine.

The corresponding mole fractions of the fullerene derivatives in the product mixtures are displayed in a box plot in Fig. 3.6. The size of each box represents the spread in the mole fractions obtained from the repeated measurements of each sultine batch. These indicate that while all sultine batches had narrow ranges of mole fractions for each adduct, the ranges were larger in Batches 3 and 4 for X₀ and X₂ and Batches 2-4 for X₁ and X₃. This can also be seen from the standard deviations of the measurements for each batch, which are shown in Table 3.1, along with the mean mole fraction of the adducts obtained from different sultine batches. For X₀ the standard deviations were 0.002, 0.001, 0.004, 0.002, and 0.003 for sultine batches one to five respectively. For X₁ they were 0.001, 0.004, 0.005, 0.005, and 0.001, for X₂ they were 0.002, 0.001, 0.004, 0.004, and 0.002, and for X₃ they were 0.001, 0.002, 0.003, 0.004, and 0.001. The overall mean mole fractions for the adducts have been shown on the plot as dashed black lines, and the standard deviations as shaded black areas. These indicate that for all adducts, the mole fractions obtained from most of the sultine batches lay within one standard deviation of the mean value. Only the measurements using Batch 1 fell outside of this range for all adducts.

To quantify the differences in the product composition between batches and within batches, ANOVA was carried out on the results. The between-batch MSEs were 0.0011, 0.0003, 0.0017, and 0.0001 for X₀, X₁, X₂, and X₃. The within-batch MSEs were 7.0×10^{-6} , 1.5×10^{-6} , 8×10^{-6} , and 6×10^{-6} respectively. Already we can see that the within-batch MSE was much lower than the between-batch MSE, strongly suggesting that the between-batch variance in the reactor product composition was greater than the within-batch variance. All of the corresponding p-values were much lower than 0.001, and therefore the differences between the within-batch and the between-batch variances were statistically significant for all adducts with a confidence greater than 99.99 %. As such, there was a

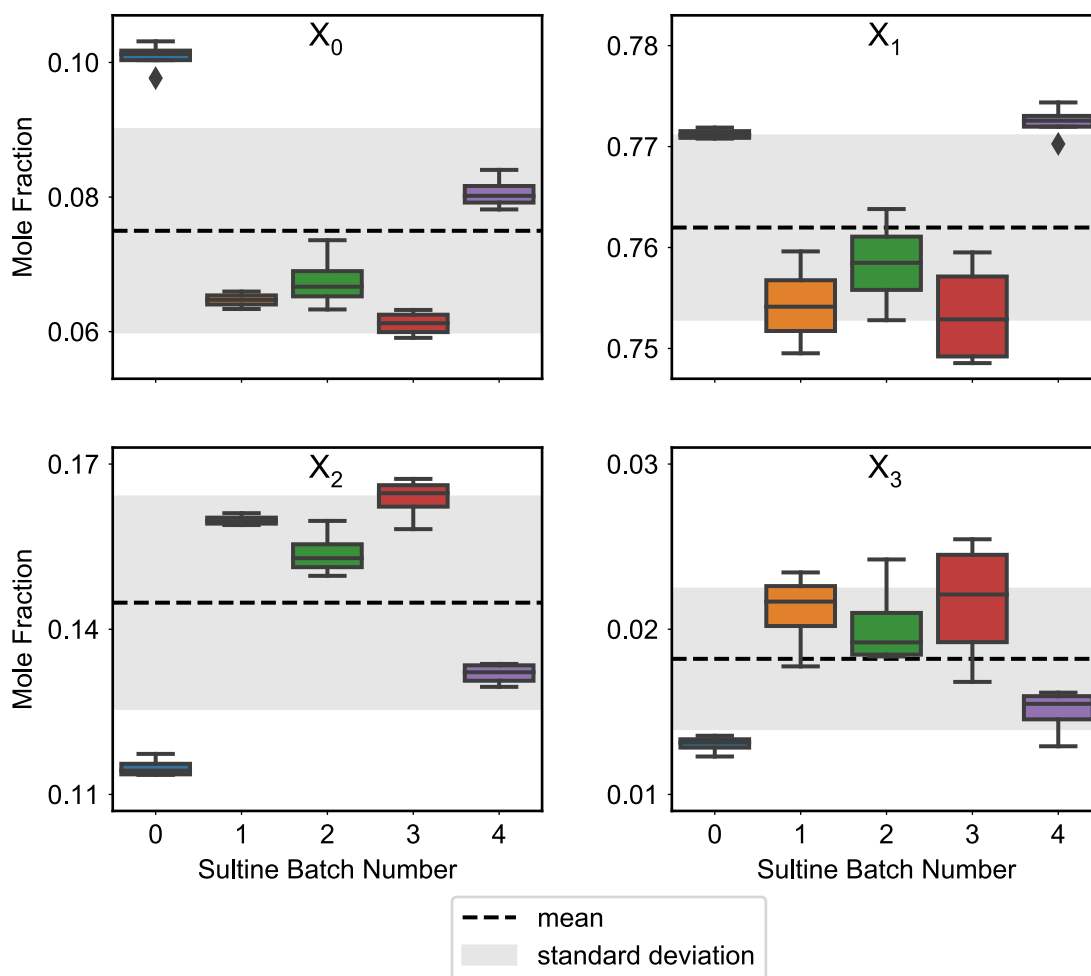


Figure 3.6: Box plots showing the spread of the mole fractions of each of the adducts in the product mixture obtained under the same reaction conditions (temperature = 111 °C, residence time = 6.2 min, [Sultine]:[C₆₀] = 5.2:1), using five different batches of sultine and four repetitions of each batch. The overall mean mole fraction of each adduct has been indicated by the dashed black lines and the standard deviation by the black shaded areas.

demonstrable increase in the variance of the product composition made under the same reaction conditions when the batch of sultine used in the synthesis was changed.

As a result of this, larger batches of sultine were synthesised to make enough stock solution to be able to run complete experiments from a single batch - even for very large experiments. However, it was impractical to make enough stock solution to run all experiments using a single batch. Despite this, the errors calculated above were all very low, even between batches of sultine. This, coupled with the results of the previous experiment, demonstrated that the automated flow reactor described here provided a stable and reliable platform for running large experiments that would be time intensive if carried out using traditional batch chemistry.

Sultine Batch	Purity / %	Mole Fraction			
		X_0	X_1	X_2	X_3
1	87.5	0.101 (± 0.002)	0.771 (± 0.001)	0.115 (± 0.002)	0.013 (± 0.001)
2	91.5	0.065 (± 0.001)	0.754 (± 0.004)	0.160 (± 0.001)	0.021 (± 0.002)
3	86.4	0.068 (± 0.004)	0.758 (± 0.005)	0.154 (± 0.004)	0.020 (± 0.003)
4	82.8	0.061 (± 0.002)	0.753 (± 0.005)	0.164 (± 0.004)	0.022 (± 0.004)
5	74.5	0.081 (± 0.003)	0.772 (± 0.002)	0.132 (± 0.002)	0.015 (± 0.001)

Table 3.1: Average mole fraction of the reactor product synthesised under the same conditions (temperature = 111 °C, residence time = 6.2 min, [Sultine]:[C₆₀] = 5.2:1) for each of the five different batches of sultine.

3.3.2 Automated Reaction Condition Screening

To demonstrate the ability of the automated flow reactor to screen reaction conditions, an experiment was carried out in which the reaction conditions were varied systematically. In this experiment, each of the reaction conditions of interest - residence time, temperature, and molar ratio of reagents - was varied in turn while holding the others at a constant value. The temperature was varied from 100 °C to 150 °C, residence time from 3 to 31 minutes, and molar ratio of reagents from 1.5:1 to 6:1. Six points were measured for each parameter, and the measurements were carried out in a random order. The effects of varying these parameters on the measured chromatograms are shown in Fig. 3.7a(i-iii).

A similar trend was observed when varying each of the reaction parameters: the unreacted X_0 peak was the most prominent at the lowest value of the reaction parameter, but became smaller as more X_0 was consumed at higher values of the reaction parameter. Correspondingly, the peaks for X_1 and X_2 both increased as the reaction parameters were increased, with X_1 being the largest of the two for all conditions tested. For the temperature and residence time measurements, a small peak for X_3 could be seen at higher values of these reaction parameters.

The stacked area plots in Fig. 3.7b show the effects on the mole fractions of the derivatives in the product mixture. For the conditions investigated, increases in temperature, residence time, and sultine:C₆₀ ratio all resulted in an increase in the consumption of C₆₀, yielding larger amounts of higher derivatives. Furthermore, the larger areas of X_2 and X_3 in the mole fraction plot for temperature suggests that, for the range of reaction

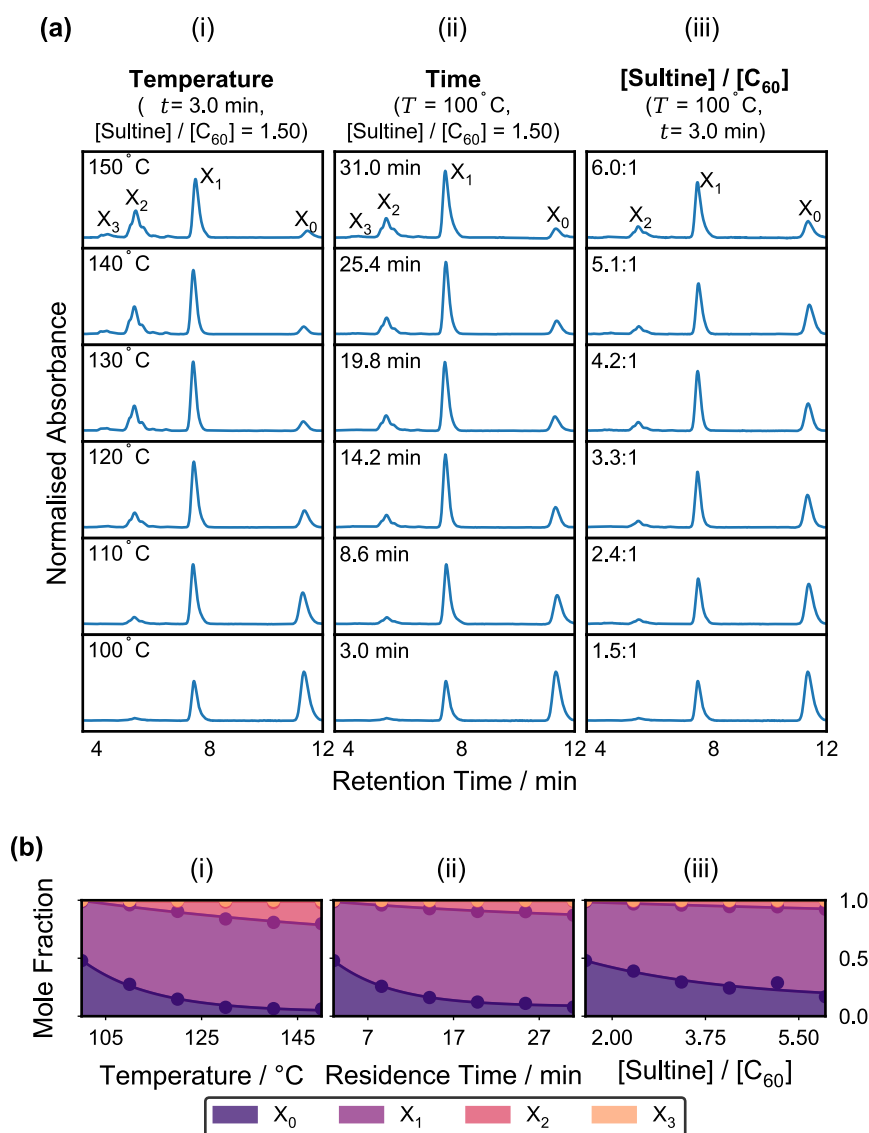


Figure 3.7: (a) Chromatograms and (b) mole fractions of the product synthesised when the temperature, residence time, and $[Sultine]:[C_{60}]$ ratio were each increased in turn, while holding the other reaction parameters constant.

conditions tested, temperature had a larger effect on the conversion to higher derivatives than the other two reaction parameters.

The smooth trends in the mole fractions of the derivatives demonstrated a well controlled reaction environment, with little noise, indicating that the flow system was a stable platform to investigate and optimise this synthesis. However, the points investigated in this experiment are only a subsection of the total parameter space. Indeed, the bounds set on the three parameters define a cube, and by varying one while the others were held constant we only investigated three of its edges. This means the plots in Fig. 3.7b only demonstrate the effect that each of the parameters had by themselves. Moving away from the edges of this cube, the combination of reaction parameters changing together could result in a more complex response surface for this reaction.

To build a more complete understanding of the relationship between the reaction parameters and the product composition, a much larger experiment was carried out to produce a coarse mapping of the whole parameter space within the defined bounds. This was done by sampling a $6 \times 6 \times 6$ grid of evenly spaced points for temperature, residence time, and reagent ratio. The compositions of the resulting product mixtures are shown as stacked plots in Fig. 3.8. The plots in each column represent a common sultine: C_{60} , while the plots in each row represent an increase in temperature from bottom to top, and the points within each plot represent increasing residence time from left to right.

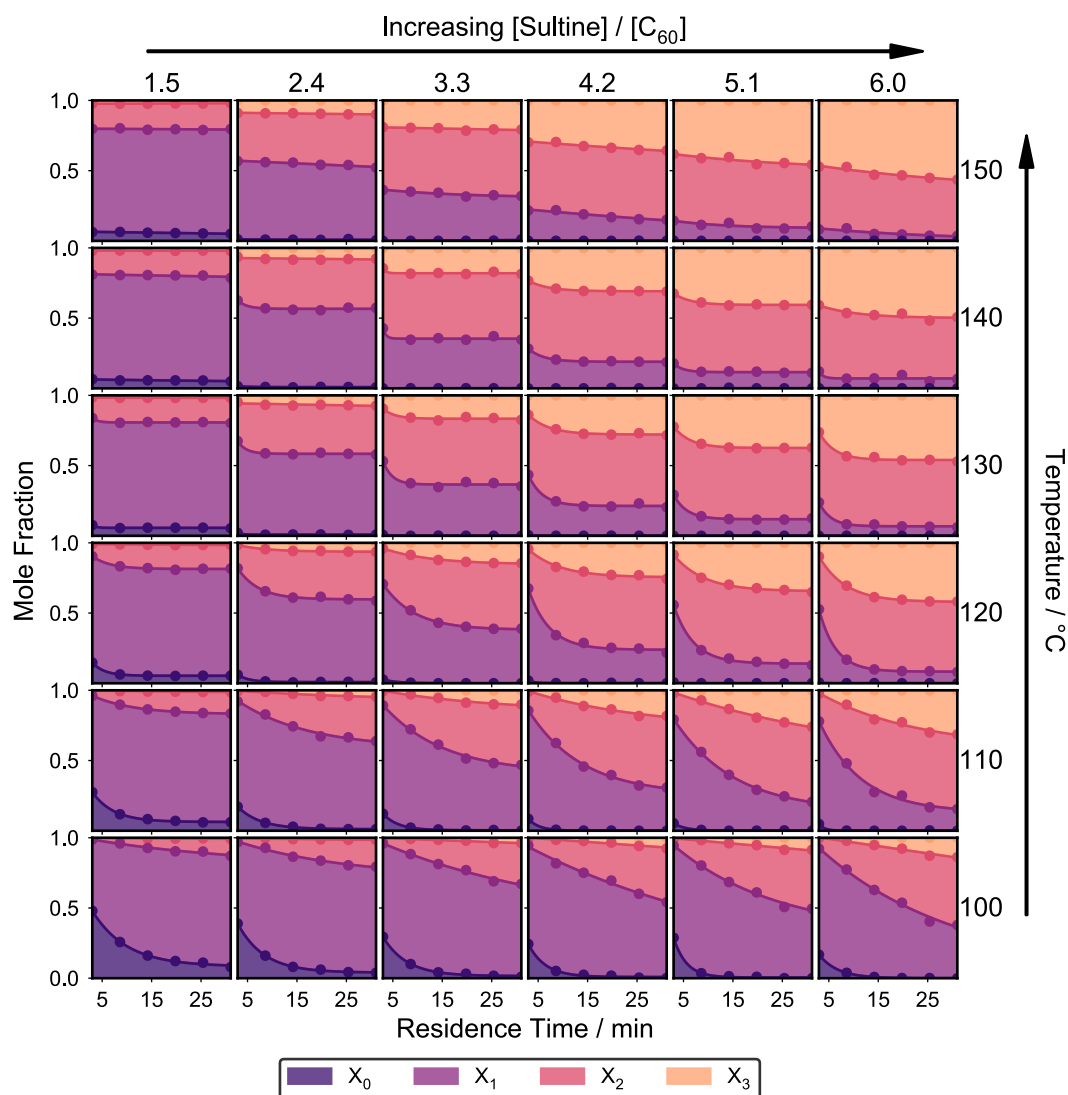


Figure 3.8: Mole fractions of fullerene adducts in products synthesised at conditions that form a $6 \times 6 \times 6$ grid in the parameter space. The temperature increases from 100 °C to 150 °C from the bottom row to the top, the $[Sultine]:[C_{60}]$ ratio increase from 1.5 to 6.0 from the left-most column to the right-most, and the residence time increase from the left of each plot to the right.

The mole fractions of X_1 , X_2 , and X_3 all increased with increasing temperature, residence time, and reagent ratio. More complex trends were observed when the reaction

parameters were increased together. As the temperature was increased and the ratio of sultine: C_{60} held constant at 1.5, higher mole fractions of X_2 were obtained at shorter residence times. For instance, the mole fraction of X_2 was 0.120 after 31 min at 100 °C, but 0.146 after just 3 min at 130 °C. Once a maximum mole fraction of X_2 was reached, around 0.180, the composition of the product did not change. This can be seen by the flattening out of the lines in the plots of the first column in Fig. 3.8. This occurred because at higher temperatures, more sultine is converted to o QDM per unit time, so larger mole fractions of the higher adducts are produced at earlier residence times. However, once all of the sultine is converted, there is no further reaction and the composition will become invariant with residence time.

Increasing the [Sultine]: $[C_{60}]$ ratio, while holding the temperature at 100 °C, caused there to be a much larger increase in the mole fraction of the higher adducts when the residence time was increased. For instance, the mole fraction of X_3 increased from 0.001 after 3 min to 0.008 after 31 min when the [Sultine]: $[C_{60}]$ ratio was 1.5, but increased from 0.005 to 0.138 when the [Sultine]: $[C_{60}]$ ratio was 6.0. This is because although the rate of conversion of sultine to o QDM will be the same, the concentration of o QDM will be higher when the [Sultine]: $[C_{60}]$ ratio is higher, thereby increasing the rate at which the higher adducts are formed.

When all three reaction parameters were increased together, the increase in the proportion of higher adducts in the product was much larger. This was because both the rate of conversion of sultine to o QDM and the concentration of sultine, and therefore o QDM, increased. This was seen in the increase in mole fractions of X_2 and X_3 from 0.012 and 0.001 at a temperature of 100 °C, a residence time of 3 min, and a [Sultine]: $[C_{60}]$ ratio of 1.5, to 0.466 and 0.472 at a temperature of 150 °C, a residence time of 31 min, and a [Sultine]: $[C_{60}]$ ratio of 6.0.

The trends in the mole fractions of the adducts were unsurprising, and each adduct had only a single maximum as their mole fractions changed with the reaction conditions. However, there was a large overlap between the mole fractions of the different adducts. This can be seen in the fact that wherever there was a large fraction of X_2 , a significant proportion of the product was X_3 . The highest mole fraction of X_2 gained over the conditions investigated was 0.534, with a mole fraction of 0.260 for X_3 at the same conditions. Similarly, where there was no X_3 , a large proportion of the product was X_1 . From the points investigated, the highest mole fraction of X_2 with less than 1 % X_3 was 0.142, corresponding to a mole fraction of 0.778 for X_1 . This all suggests that there is no combination of reaction conditions, within the bounds of the parameter space chosen here, that would give pure X_2 or X_1 .

Finally, the smooth, systematic nature of the trends in the distribution of mole fractions of the derivatives suggests once again that the complete system gave a stable and reproducible output. This is further emphasised by the fact that when the experiment

was run, the order of the points was randomised and random points were selected to be repeated throughout the experiment. The chromatograms of these repeats can be seen in Fig. 3.9. In total 36 repeats were carried out, with 5 points repeated 3 times, 20 points repeated twice, and 191 points run only once.

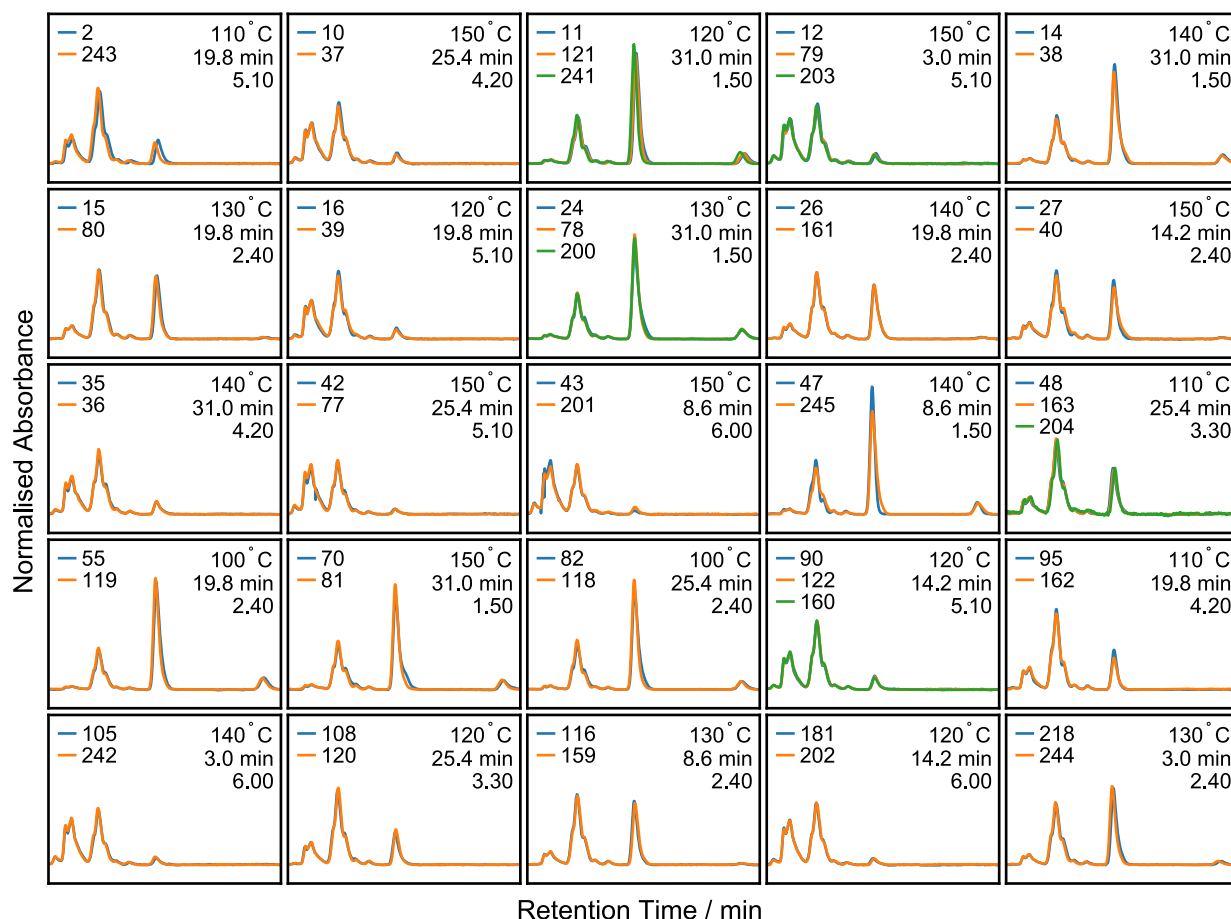


Figure 3.9: Chromatograms of measurements that were repeated during the experiment to synthesise products using reaction conditions forming a $6 \times 6 \times 6$ grid in the parameter space. The measurement number in the experiment of each chromatogram is indicated in the legend of each plot, while the reaction conditions used are shown in the top right corner.

The chromatograms of the repeated points overlap closely. The measurement numbers at which the chromatograms were acquired during the experiment are shown in the legend of each plot. Some of the repeats were carried out much later in the experiment than the original point, yet still yielded virtually the same product composition, indicating good reproducibility of the flow system. In total the experiment comprised 246 measurements, which would have required a very significant amount of labour, equipment, and/or time to carry out in batch and would likely have resulted in much poorer reproducibility. However, the only work needed here was to make up the stock solutions, load them onto the reactor, and feed the reactor the points to investigate. Despite generating a relatively large amount of data, the experiment only took four days to run.

3.4 Self-Optimisation

A large number of problems in synthetic chemistry come down to optimisations. Often the aim is to maximise the yield of a target molecule, while at the same time minimising any unwanted side products, or to optimise a particular property of a material, like making a polymer with a particular average molecular weight or a nanoparticle with a particular average diameter. The increased control of reaction conditions makes flow systems ideal platforms for carrying out chemical optimisations. Furthermore, the modular nature of flow systems simplifies automation, as well as allowing for incorporation of in-line analysis of the product-outflow. This, coupled with their typically small volumes, makes it possible to carry out experiments in which a large number of reaction parameters are varied, and a product analysed at each set of conditions - something that is necessary to efficiently run an optimisation.

Automated flow reactors have been used for the self-optimisation a handful of syntheses. Most of these focus on optimising a single target^{78–84,87,89}, typically the yield of a synthesis, although there are examples of what could be classed as multi-objective optimisations, where the focus is optimising two or more possibly competing targets^{88–90}. These multiple targets can be reaction outcomes, such as balancing production rate with conversion efficiency⁸⁹, or physical properties, such as maximising the intensity of quantum dot emission while tuning it to a target wavelength⁹⁰.

Multi-objective optimisation of a chemical synthesis often means achieving an acceptable trade-off between competing criteria. For example, longer residence times typically promote increased yield of a product but slow down how fast that product is made. Optimising both yield and production rate therefore requires balancing increasing one at the cost of the other. Our example synthesis of *o*-xylenyl derivatised fullerenes is an example of a multi-objective optimisation that requires optimisation of a trade-off between maximising production of the doubly-substituted derivative (X_2) and suppressing production of the detrimental triply-substituted derivative (X_3).

3.4.1 The Choice of Optimisation Algorithm

A chemical synthesis can be viewed as a black-box optimisation, where the underlying functions that generate the measurable output are unknown. The underlying function may have multiple optima, which can only be known by evaluating the ‘black-box’ function at different points in the parameter space. However, running a chemical synthesis can be expensive in terms of both the cost of the materials, and the time it takes to run it. As such, a self-optimisation of a chemical synthesis needs to balance the aim of converging to an optimum in as few evaluations as possible with the need for exploring the parameter space enough to not get stuck in any sub-optimal local minima.

Local optimisation routines avoid the need to worry about exploring the full parameter space, by accepting that they may not find the global optimum, and assuming the optimum in a particular region of the parameter space is good enough. Many of the examples of chemical self-optimisations in the literature prefer so-called direct search algorithms^{72,78,79,81,83,84,88,89,107}, such as the Nelder-Mead Simplex algorithm, as they do not need to calculate any derivatives to find the direction to search in, and so require fewer function evaluations. It has been posited that most chemical syntheses exhibit very simple surfaces relating the reaction conditions to their output⁹⁶. In this case, local optimisations may be preferred as they could lead to faster identification of an optimum.

However, in the case of a multi-objective optimisation where multiple objectives are competing with each other, the parameter space can become complex, as the trade-offs between the objectives interact in unexpected ways. As seen in the previous section, the competition between consumption and production of each of the different products in the synthesis of *o*-xylenyl derivatised fullerenes leads to multiple points in the parameter space yielding products mixtures with similar compositions. In this case, a global optimisation algorithm is necessary.

While there are a number of different global optimisation routines, so far the only one that has been used for chemical self-optimisations in the literature is the Stable Noisy Optimisation by Branch and Fit^{21,80,81,87} (SNOBFit), developed by Huyer and Neumaier⁹⁵. This algorithm balances evaluating points around minima that have already been found, by performing local fits, with looking for other minima that may also exist, by evaluating points in unexplored areas of the parameter space.

There are also a number of other features that made SNOBFit a good choice of algorithm for the self-optimisations carried out in this chapter. As mentioned previously, the algorithm recommends batches of points to evaluate, the size of which can be chosen by the experimenter. This allows syntheses to be run in parallel. Although the flow system developed in this chapter could only sample points sequentially, it was set up to run batches of points efficiently by running the synthesis of one point while the analysis of the previous one was carried out. Similarly, the algorithm is tolerant of noise in the objective function that it is minimising, as long as the noise is stable and does not change during the optimisation. For these reasons, SNOBFit was chosen as the algorithm to use for optimising the synthesis of *o*-xylenyl derivatised fullerenes.

3.4.2 2D optimisation

To test whether the flow system would perform well using SNOBFit for a self-optimisation, a initial two-dimensional optimisation was run. While an obvious target for the optimisation might be to maximise the amount of X₂ in the product, it has already been shown in the previous section that this would lead to a result unsatisfactorily high in X₃. As such,

a combined objective function was used in an attempt to optimise the trade-off between maximising X_2 and minimising X_3 . The objective function (Eqn. 3.1) was formulated as a minimisation, that decreased (became more optimal) with increasing $[X_2]$ and increased (became less optimal) with increasing $[X_3]$. It was also scaled to have a maximum of 1 and a minimum of 0.

$$f(\mathbf{x}) = \frac{1 - [X_2] + [X_3]}{2} \quad (3.1)$$

The temperature of the reactor was held fixed at 120 °C, while the maximum overall flow rate was bounded between 61 μLmin^{-1} and 922 μLmin^{-1} , and the ratio between the reagent flow rates was bounded between 0.25 and 4. This allowed the residence time to vary between 1.0 and 15.1 min, and the molar excess of sultine to vary between 0.75 and 12.

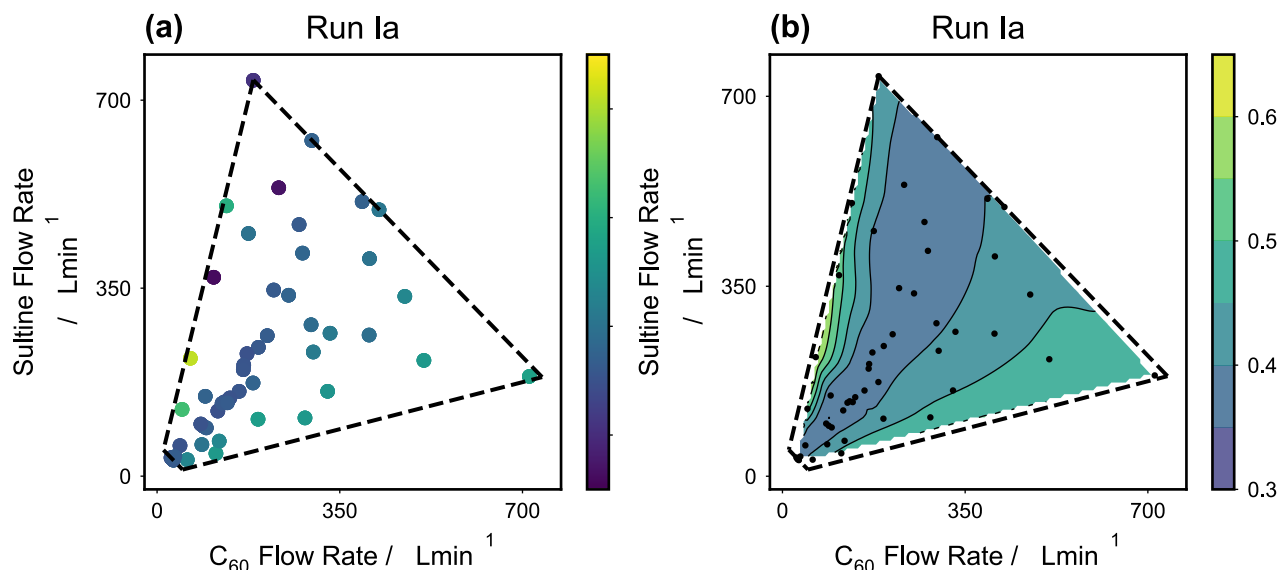


Figure 3.10: Results of the first trial SNOBFit optimisation of the synthesis of *o*-xylenyl derivatised fullerenes, Run Ia, as (a) a scatter plot and (b) a contour plot of a surface fit to the results (b). The optimisation was 2-dimensional, varying the overall flow rate between 61 μLmin^{-1} and 922 μLmin^{-1} and the ratio between the reagent flow rates between 0.25 and 4, while keeping the temperature fixed at 120 °C. It was run for a total of 54 measurements.

The optimisation was run for 54 points, with 6 random starting points, and 6 points being recommended by each call to the SNOBFit algorithm. The results from this run can be seen in Fig. 3.10a, where a scatter plot shows the points in the parameter space sampled during the optimisation. The points have been coloured according their value of $f(\mathbf{x})$, with darker points having lower (i.e. better) values. The bounds of the optimisation are shown by a black dotted line. The points in the scatter plot appear to form a channel along the middle of the plot, clustered towards lower flow rates, and with a few points sampled away from all the others towards the boundary. The points in the clustered

channel are all much darker than those sampled outside of it. This suggests that best points are found along the line defining equal flow rates of the reagents, corresponding to a molar excess of sultine of 3, with the objective function increasing as the flow rates move to larger excesses of either reagent.

This can be more clearly seen in Fig. 3.10b, where a contour plot has been used to show a cubic fit to the optimisation data, with the data points from the run being overlaid in black. Here the colour of the contours gets darker as you move in from the edges of the plot, as the $f(\mathbf{x})$ decreases. A large swathe of light purple can be seen in the middle of the plot, forming a channel along the line defining equal reagent flow rates. Although this suggests that low values of the objective function can be found at any of the flow rates within the bounds chosen for this optimisation, the best overall point found during the optimisation was at slower flow rates. The best function value found was 0.395, at an overall flow rate of $180.8 \mu\text{Lmin}^{-1}$ and a ratio of flow rates of 0.86, giving a residence time of 5.1 min and a sultine excess of 2.6.

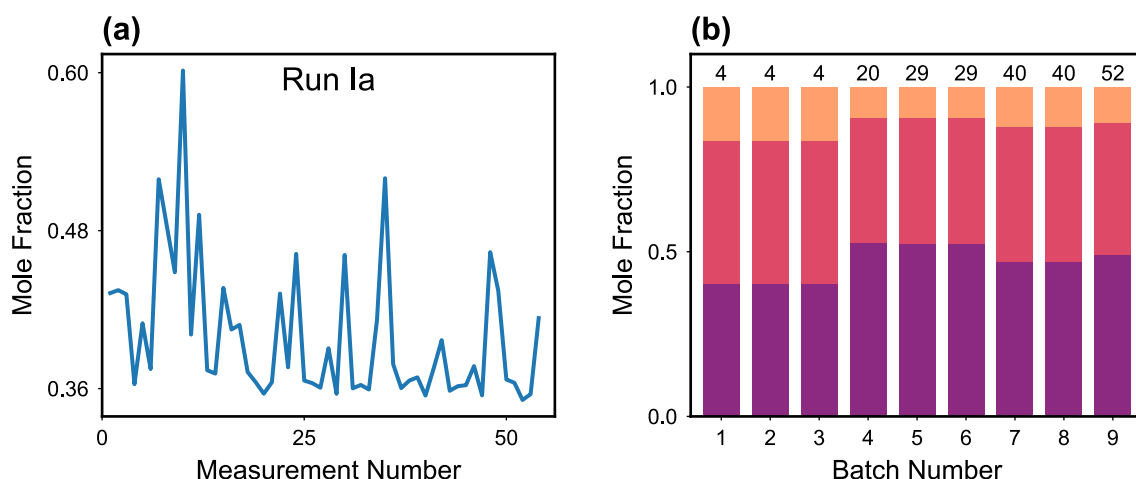


Figure 3.11: (a) The value of the objective function, Eqn. 3.1, at each measurement in Run 1a and (b) the composition of the best product mixture identified from each batch of points during the optimisation, with the measurement number of the best point indicated above each bar.

In Fig. 3.11a the measured objective function value is plotted against the measurement number, showing how the optimisation progressed during the run. This does not show a lowering of the function value with each consecutive point, as might be expected, but shows a fluctuation between low and high $f(\mathbf{x})$ values. This demonstrates the local and global searching modes of the SNOBfit algorithm. Each time the SNOBfit function is called, it recommends a batch of points that is a mixture of expected minimisers for local fits around previously found promising points, and points that are in unexplored parts of the parameter space. The almost periodic rise and fall of the function value seen in Fig. 3.11a is due to the local search occurring first in each batch, before the exploratory search at the end.

The improvement in the product composition throughout the optimisation is shown in Fig. 3.11b. Each stacked bar shows the composition of the current best point found at the end of each batch of recommended points. The mole fraction of the undesirable derivative X_3 started at 0.161 at measurement number 4, before dropping to 0.092 at measurement 20. It rose again to 0.094 then 0.120 at measurement 29 and 40, before ultimately dropping to 0.106 at measurement 51, the best result in this optimisation. The mole fraction of X_2 was 0.435, 0.380, 0.382, 0.410, and 0.404 at the same points. This shows the optimisation quickly identified a good point rich in X_2 , but also with a relatively high amount of X_3 . As the optimisation progressed, it improved on the best result by balancing the amounts of X_2 and X_3 - subsequent improved results saw the mole fractions of both X_2 and X_3 decrease, before increasing.

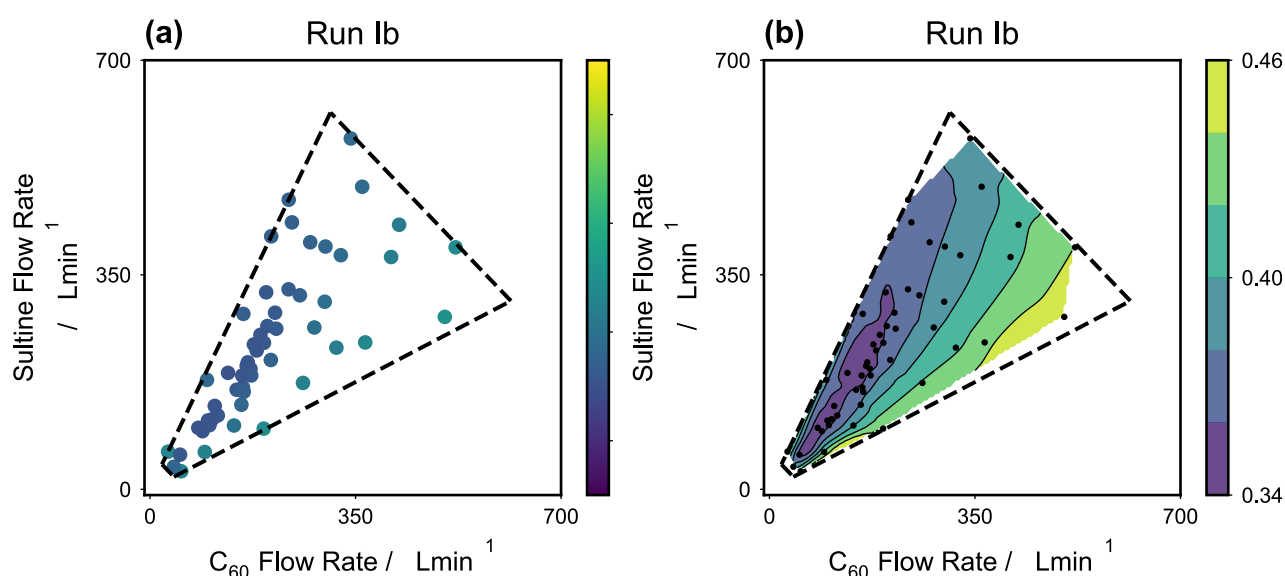


Figure 3.12: Results of the second trial SNOBFit optimisation using stricter bounds, Run Ib, as (a) a scatter plot and (b) a contour plot of a surface fit to the results. The optimisation was 2-dimensional, varying the overall flow rate between $61 \mu\text{Lmin}^{-1}$ and $922 \mu\text{Lmin}^{-1}$ and the ratio between the reagent flow rates between 0.5 and 2, while keeping the temperature fixed at 120°C . It was run for a total of 54 measurements.

To find out if a more optimum point could be found, the bounds of the the optimisation were restricted. As all of the best points were found with nearly balanced flow rates for the two reagent solutions, the maximum and minimum flow rate ratios were changed to 2 and 0.5 respectively. By narrowing the bounds of the optimisation, the size of the parameter space was reduced, meaning a greater proportion of the points should have been sampled from near the optimum point. It was hoped that this would either find the optimum with fewer objective function calls, find a more optimum point, or both.

The results from this second optimisation run can be seen in Fig. 3.12a. This scatter plot looks similar to the previous run, but with a tighter cluster of points at lower flow rates. The contour plot in Fig. 3.12b confirms what was expected: by narrowing the

bounds of the optimisation, a greater number of near optimal points were found. The best point found in this optimisation had an objective function value of 0.351 at measurement number 23, with an overall flow rate of $322.9 \mu\text{Lmin}^{-1}$ and a flow rate ratio of 1.43. This translates to mole fractions of 0.000, 0.381, 0.459, and 0.160 for X_0 to X_3 , at a residence time of 2.9 min and a sulfate excess of 4.3.

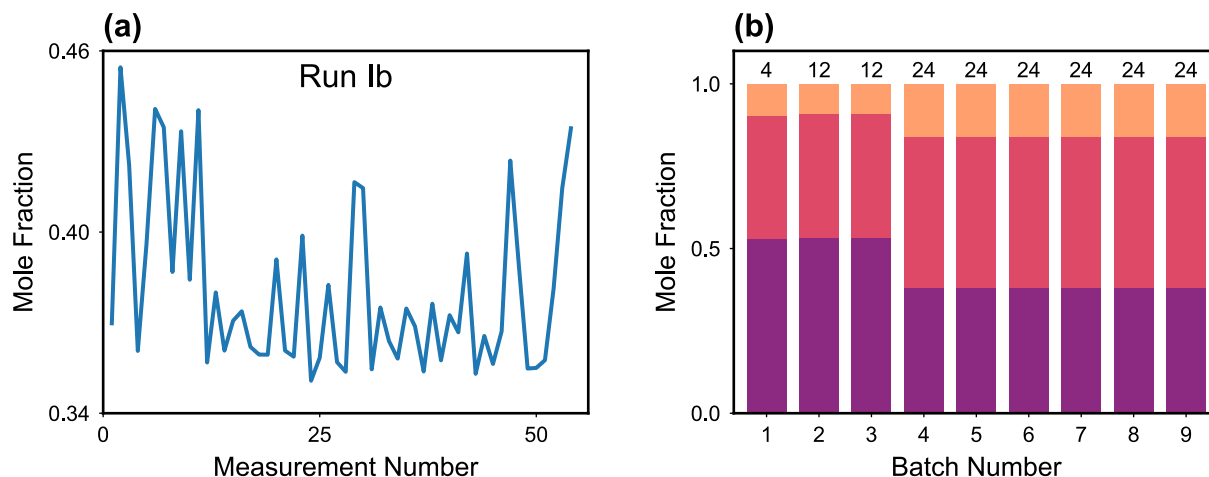


Figure 3.13: (a) The value of the objective function, Eqn. 3.1, at each measurement in Run Ib and (b) the composition of the best product mixture identified from each batch of points during the optimisation with the measurement number of the best point indicated above each bar.

The plot of $f(\mathbf{x})$ against measurement number in Fig. 3.13a looks much the same as in the previous run, with the exception that the range of values is reduced - the minimum being 0.351 and the maximum 0.455 compared to 0.351 and 0.602 for the previous run. Although both optimisations found an optimum with the same value of $f(\mathbf{x})$, Fig. 3.13b demonstrates that running the optimisation with reduced bounds lead to it converging on the minimum much sooner. In fact, the optimum value of 0.351 was found in less than half the number of function calls as before - at measurement number 23 compared to number 51 in Run Ia.

A similar progression in the product composition at the current best point in the optimisation can be seen, albeit over a shorter number of measurements. The amount of X_3 in the product at the running best points was 0.096, 0.090, and 0.160 at measurements 4, 12, and 24, while the amount of X_2 in the product at these same points was 0.375, 0.377, and 0.459.

The composition of the best points for each optimisation run is compared in Fig. 3.14a, more clearly illustrating that although the first run had a lower amount of X_2 in its product, it also had a lower amount of undesirable X_3 . This highlights what was a problem with the formulation of the objective function: product mixtures with different compositions led to equal function values as long as any increase in X_3 were accompanied by an equal increase in the amount of X_2 .

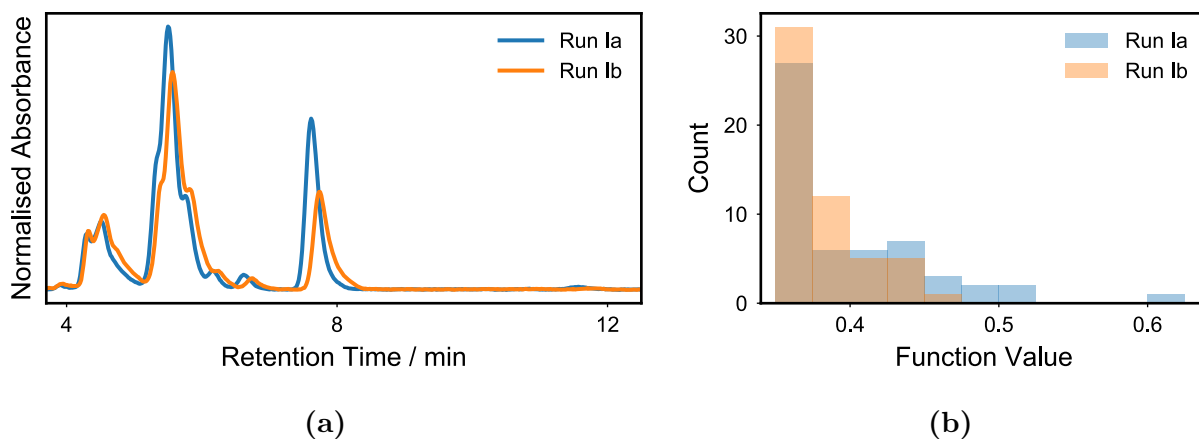


Figure 3.14: The best results from Run Ia and Run Ib shown as (a) the measured chromatograms and (b) the distribution of function values, of Eqn. 3.1, for both optimisations.

In Fig. 3.14b, the distribution of objective function values sampled during each run of the optimisation has been plotted. Both optimisations show distributions of function values that are skewed towards the lowest function value. This demonstrates that the SNOBfit algorithm is working as expected. Also as expected, narrowing the bounds in the second optimisation produced a narrower distribution as a result of the reduced size of the parameter space available to sample from. If both optimisations were run for longer, the distribution of function values would tighten, and the two distributions would be expected to converge.

Through this experiment, it was demonstrated that the reactor was capable of running self-optimisations for at least 50 points. From running two optimisations with different bounds it appears that although the algorithm is able to easily find the lowest function value, it is important to choose the bounds carefully for it to converge quickly.

3.4.3 3D optimisation

After running successful 2D optimisations, 3-dimensional optimisations were carried out to vary all three of the reaction conditions that the automated flow reactor could control. The same objective function as before (Eqn. 3.1) was used to optimise the trade-off between desired increases in X_2 against unwanted increases in the amount of X_3 . An optimisation was carried out with the overall flow rate bounded between 92 and 922 μL^{-1} , meaning the residence time was limited between 10 and 1 min, and the flow rate ratio was bounded between 2 and 0.5, meaning the molar excess of sultine was limited to between 6 and 1.5 times the molar concentration of C_{60} . The temperature was bounded between 100 and 150 $^{\circ}\text{C}$, and the optimisation was run for 105 function evaluations, i.e. 14 calls to the SNOBfit algorithm.

The scatter plot in Fig. 3.15a shows the points that were investigated during this optimisation run. Like the previous 2D optimisations there is a tight cluster of points,

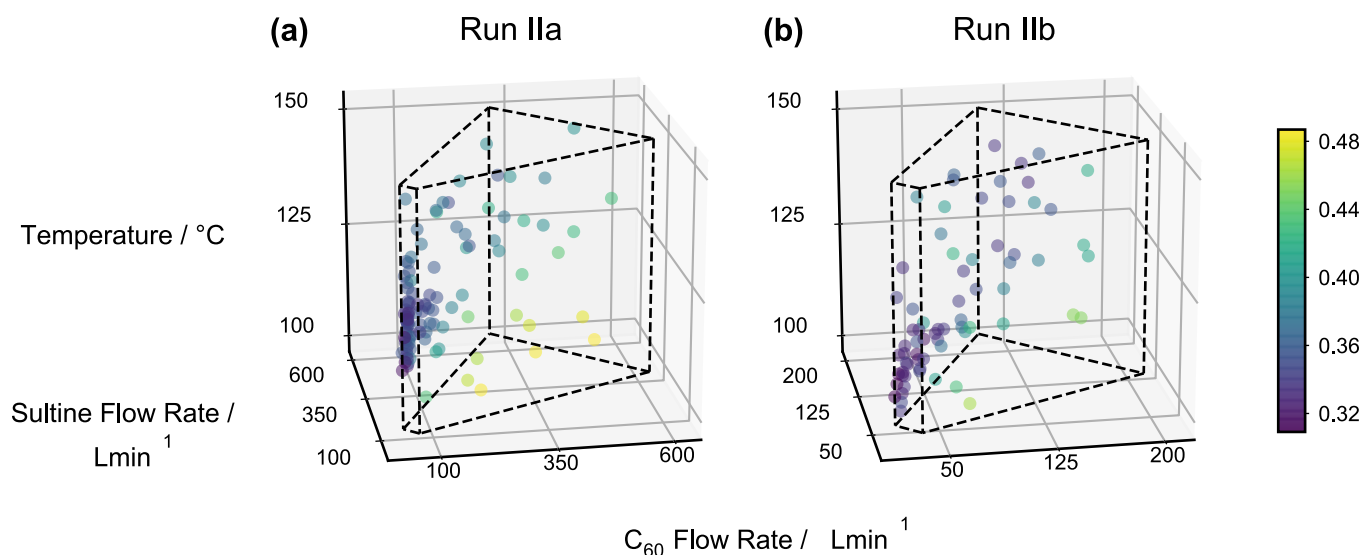


Figure 3.15: Comparison of the points sampled from two 3D optimisations with two different bounds, (a) Run IIa and (b) Run IIb, shown as scatter plots with the colour of the points representing the value of the objective function, Eqn. 3.1. In Run IIa the bounds were 922 and 92 μL^{-1} for the overall flow rate, 2 and 0.5 for the ratio between the flow rates, and 150 °C and 100 °C for temperature. In Run IIb the bounds were 310 and 50 μLmin^{-1} for the overall flow rate, 2 and 0.5 for the ratio between the flow rates, and 150 °C and 100 °C for temperature. Run IIa was run for 105 measurements, while Run IIb was run for 70.

as well as a small number of other points spread throughout the parameter space. The best result was found at measurement number 98, with a function value of 0.315. The composition at this point was 0.000, 0.373, 0.496, and 0.131 for $[X_0]$, $[X_1]$, $[X_2]$, and $[X_3]$ respectively. It is notable that a lower function value of 0.315 (compared to 0.351 for both previous runs), was achieved by allowing the temperature to vary. Indeed, this improved optimum was found at a lower temperature of 114 °C than the 120 °C used before. All of this showed that the reactor could successfully run full 3D self-optimisations.

However, most of the points in Fig. 3.15a appear clustered up against the low-flow rate boundary, at the ‘spine’ of the trapezoidal prism formed by the flow-rate boundaries. This suggests that the boundaries of the optimisation were poorly chosen, and a more optimum set of conditions might have been found at lower flow rates, or longer residence times. Furthermore, the best point in the optimisation was not found until near the end, at measurement number 98, meaning the optimisation may benefit from changing the bounds to reduce the overall volume of the parameter space.

A second optimisation was therefore run, but with the bounds on the overall flow rate changed to 50 and 310 μLmin^{-1} , giving maximum and minimum residence times of 18.4 min and 3.0 min respectively. The bounds for the flow rate ratio and temperature were kept the same, and the optimisation was run for a reduced number of 70 measurements, or 9 calls to the SNOBFit algorithm. The points investigated during this run are shown in Fig. 3.15b. Again, most of the points with the lowest function values were clustered

at low flow rates and lower temperatures. This time, the best point was found after 57 measurements with a slightly better (lower) value of 0.309, at a residence time of 8.8 min, a molar excess of sultine of 6.0, and a temperature of 116 °C. Although the best point of this optimisation run was at a higher temperature than before, it is clear that by reducing the maximum allowed residence time to 18.4 min, a better minimum was found. Furthermore, as seen with the 2D optimisations, by reducing the size of the parameter space to be investigated, a better point than before was found with fewer function evaluations.

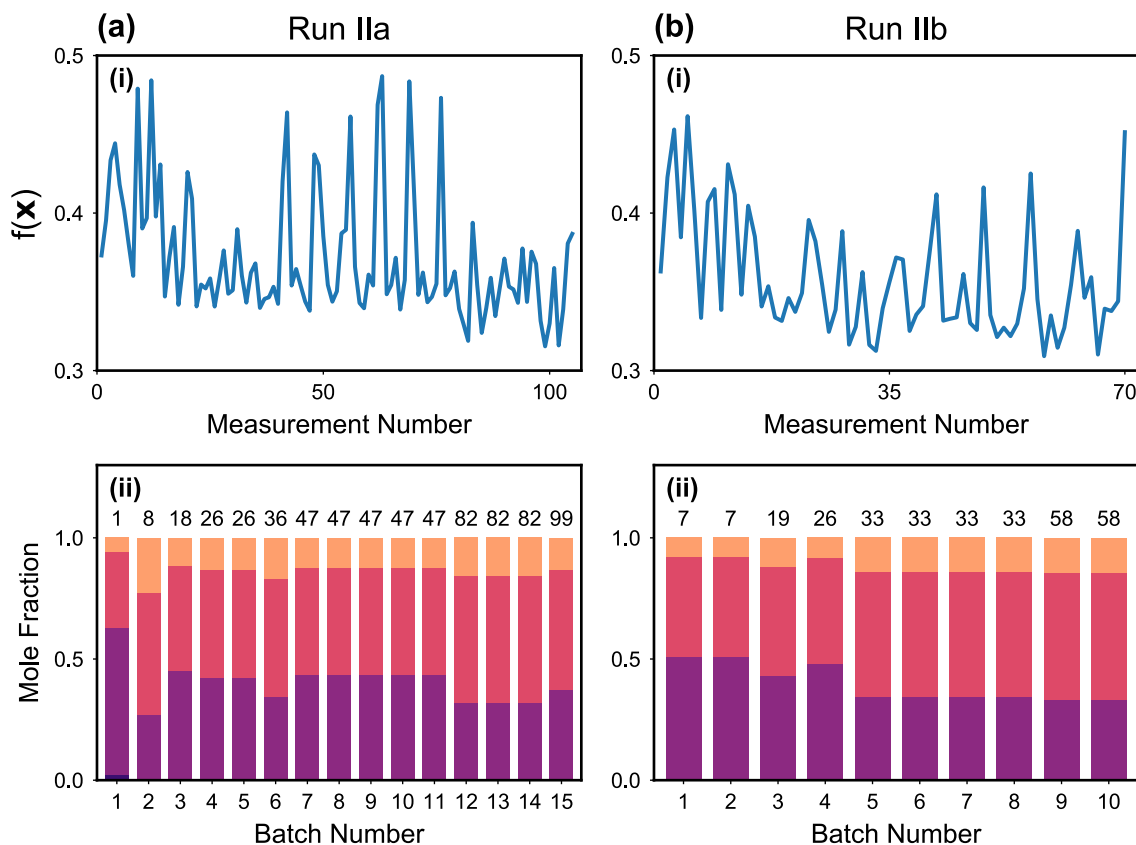


Figure 3.16: (a) The value of the objective function, Eqn. 3.1, at each measurement in (i) Run IIa and (ii) Run IIb. (b) The composition of the best product mixture identified from each batch of points during (i) Run IIa and (ii) Run IIb, with the measurement number of the best point indicated above each bar.

The improvement in the convergence of the optimisation by changing the bounds of the optimisation can be seen in Fig. 3.16a(i) and Fig. 3.16b(i), where the function value is plotted for each consecutive measurement. Here, the range of function values sampled was higher in the wider optimisation (Run IIa, Fig. 3.16a(i)) than for the narrower optimisation (Run IIb, Fig. 3.16b(ii)) - the range of the former being between 0.351 and 0.487 while that of the latter was 0.309 and 0.462. This suggests that more of the points sampled during the second optimisation were near the optimum, and that the points sampled during the local search of each SNOBFit call were close to the overall minimum.

This is backed up by the plot of product composition at the running best point. For Run IIa, the so we can see that the composition changes in Fig. 3.16a(ii), due to the current best point changing with almost every call to the SNOBFit algorithm. There is no obvious trend in the mole fractions of each of the fullerene derivatives, apart from X_0 which decreases rapidly to zero. As well as suggesting this optimisation has not converged yet, it also highlights the fact that a product with less X_2 than another can have a higher objective function value if the amount of X_3 is also lower.

Improved convergence is also suggested by how the running best point changed during each optimisation. For Run IIa (Fig. 3.16a), the running best point changed regularly, accompanied by relatively large changes in composition. In comparison, the running best point for Run IIb (Fig. 3.16b(ii)) also changed regularly at first, but after the first four batches of points remained relatively stable. The optimum point was found at measurement number 57, compared to 98 for Run IIa, with mole fractions of 0.000, 0.345, 0.521, and 0.145 for X_0 , X_1 , X_2 , and X_3 respectively. This showed an improvement in the amount of X_2 , from 0.496 to 0.521, at the cost of a slight increase in the amount of unwanted X_3 , from 0.131 to 0.145. As a result, there was an improvement of just 0.006 in the objective function.

The chromatograms of the best result found by both 3D optimisations are compared in Fig. 3.17a. As the mole fractions quoted above show, both had very similar compositions. However, the best result of the second optimisation had slightly less X_1 , and more X_2 , so had a lower objective function value.

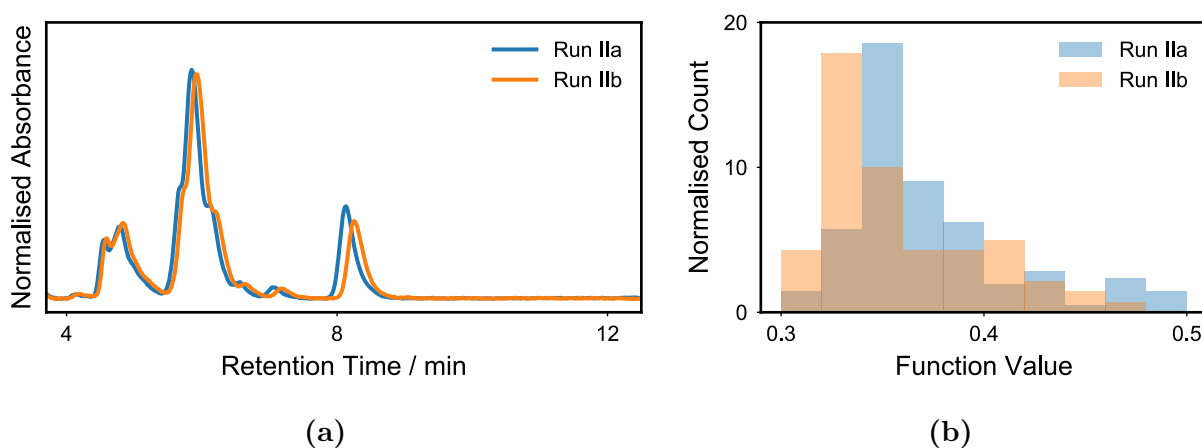


Figure 3.17: Comparison of the best results from Run IIa and Run IIb shown as (a) the measured chromatograms and (b) the distribution of function values, of Eqn. 3.1, for both optimisations.

As well as demonstrating that the reactor could successfully perform 3D self-optimisations, these two experiments further highlight the need to choose appropriate bounds for the input parameters of an optimisation. In the previous 2D optimisations, changing the bounds on the parameters allowed the optimum point to be found faster. In these experiments,

the first optimisation became stuck at the parameter boundary and changing the bounds allowed a better optimum to be found. In both, the optimum point was found with fewer evaluations of the objective function and more points were sampled near this optimum, after the bounds on the parameters were changed. This is reflected in the distribution of function values sampled during Run IIb having a larger skew to lower values than the distribution for points sampled during Run IIa (Fig. 3.17b).

However, these results also suggest that the objective function needs careful formulation before starting an optimisation. The objective function here gave equal weighting to the amount of X_2 and the amount of X_3 in the product. As such, an increase in the amount of X_3 could result in a lower function value, as long as it was accompanied by a slightly larger increase in the amount of X_2 . This may be fine, if the goal is to maximise the yield of X_2 . But from an electronic device perspective, the tertiary product is the least wanted and the most detrimental to the application of these molecules. Therefore, a different objective function may be required to better reflect the requirements of the product.

3.4.4 Changing Objective

All of the previous runs used a merit function that rewarded production of X_2 and penalised production of X_3 . However, all gave an optimised trade-off that was too high in X_3 . This could be remedied by adding a large weighting coefficient to the $[X_3]$ term in Eqn. 3.1, to more heavily penalise it, but it is not obvious what value to choose. Results in the previous section showed a large overlap between the production of X_2 and its consumption to form X_3 . Penalising $[X_3]$ too much, i.e. using too large a weight, could push the product towards compositions with very low $[X_2]$. On the other hand, being too lenient would lead to good levels of $[X_2]$, but too much $[X_3]$. In fact, the best results from the optimisations in the previous sections had similar compositions to the point with the most X_2 found during a systematic search of the parameter space. This suggests that optimising Eqn. 3.1 the same as not penalising production of X_3 at all.

To confirm this, and also confirm that optimising just $[X_2]$ was not a viable strategy, an optimisation to maximise the amount of X_2 in the product was run, using in the objective function in Eqn. 3.2.

$$f(\mathbf{x}) = 1 - [X_2] \equiv [X_0] + [X_1] + [X_3] \quad (3.2)$$

The bounds were chosen using the parameter space search from the previous chapter to ensure the maximum would be around the centre of the parameter space, with temperature restricted to between 120 and 180 °C, the overall flow rate between 30 and 184 μLmin^{-1} , and the flow rate ratio between 0.5 and 2. This gave residence times between 30.7 and 5.0 min, and molar excesses of sultine of between 1.5 and 6. The optimisation was run

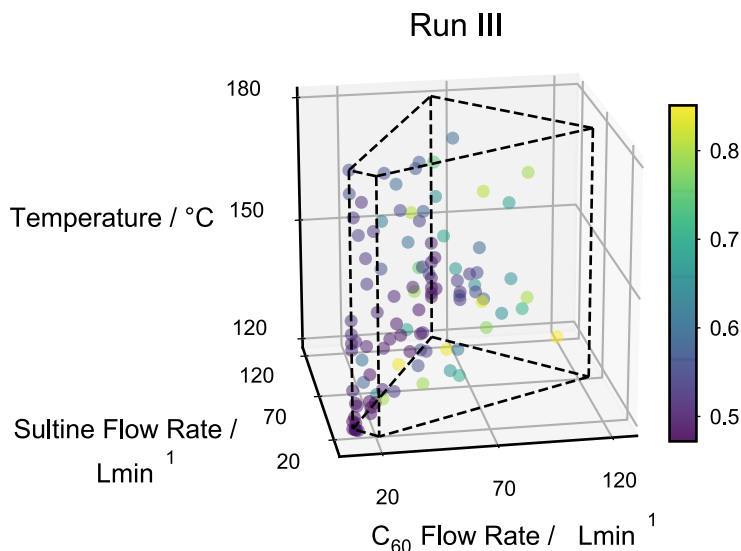


Figure 3.18: Scatter plot of results from the SNOBFit optimisation targeting maximum $[X_2]$, Run III, as a scatter plot, where the colour of points represents the value of the objective function, Eqn. 3.2. The bounds used were 30 and 184 μLmin^{-1} for the overall flow rate, 0.5 and 2 for the reagent flow rate ratios, and 120 and 180 $^{\circ}\text{C}$ for the temperature

for 105 measurements.

The scatter plot of the results in Fig. 3.18 shows that the points with the lowest function value, and therefore most X_2 , were clustered towards lower temperatures and flow rates. The overall minimum point was found at measurement number 75 with a residence time of 5.0 min, a molar sultine excess of 4.6, and a temperature of 133 $^{\circ}\text{C}$. The objective function value was 0.472, giving the mole fraction of X_2 as 0.528. This was slightly higher than the 0.521 achieved by the previous optimisations.

The plot in Fig. 3.19a shows how the composition of the product changed as successively better results were found. Early improvements showed a decrease in both the amount of X_1 and X_3 , starting at 0.189 and 0.334 at measurement 3, and then decreasing to 0.203 and 0.298, then 0.170 and 0.303 at measurements 11 and 15. The final improvement to the optimum point at measurement 75 gave a further reduction in X_3 to 0.295, but a slight increase in X_1 to 0.176.

However, there was much more X_3 present at the best point than with the previous objective function: the mole fraction of X_3 was 0.295, making it almost a third of the product. The amount of X_0 was still negligible, but the increase in X_3 resulted in a decrease in X_1 to 0.176. This showed including a penalty on the production of X_3 (as in Eqn. 3.1) did yield a product with less X_3 , without significantly reducing the amount of X_2 .

Despite this, these experiments confirmed that targeting a maximum amount of X_2 was not a viable strategy in this synthesis, and that the penalty imposed upon the amount of X_3 in the product during previous optimisations was not large enough. While further

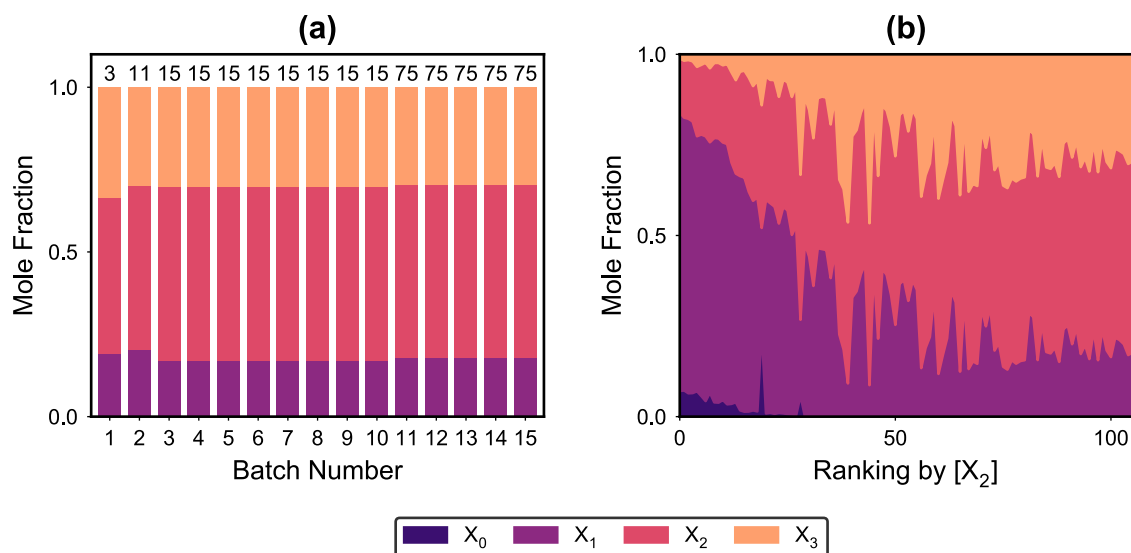


Figure 3.19: (a) The composition of the best product mixture identified from each batch of points during the SNOBFit optimisation in Run III with the measurement number of the best point indicated above each bar and (b) the composition of all products measured during Run III, ranked by their mole fraction of X_2 from the lowest at 0 to the highest at 105.

experiments could have been run to determine the right penalty, the results gained so far suggested this wasn't a good idea. The composition of products sampled during run III are shown in Fig. 3.19b ranked by their mole fraction of X_2 . This shows that decreases in the amount of X_1 were accompanied by an increase in the amount of X_2 , and a corresponding increase in the amount of X_3 . There appears to have been a slight decrease in $[X_3]$ in the product composition after the initial increase, but all compositions that were high in X_2 also had large amounts of X_3 . This confirms that the scope for optimising this trade-off may have been limited, and an alternative approach was needed.

3.5 Summary

This chapter described work carried out to develop a fully automated flow reactor for the synthesis of *o*-xylenyl functionalised fullerene derivatives. A self-built continuous flow reactor was chosen for simplicity and flexibility. The complete reactor was composed of syringe pumps that delivered reagents into a PTFE tube reaction channel, a PID controlled solid-state heater for heating the reaction channel, and an in-line HPLC system for product analysis. Linking the syringe pumps, heater, and HPLC system to a central computer allowed automated control of the reactor.

Experiments were first carried out to characterise the stability and reproducibility of the system. The product composition was found to be stable over multiple days, and although using different batches of sultine stock solutions resulted in an increase in the variability of the composition, the overall variability in the distributions of mole fractions

of the fullerene derivatives was low ($<0.5\%$).

The flow system was used to measure the effect of systematically varying the reaction parameters on the product composition. At first, each controllable reaction parameter - the temperature, the residence time, and the reagent ratio - was varied in turn while the others were held constant. The outcome was broadly as expected, with there being an increase in the mole fraction of higher derivatives produced, and an increase in conversion, as each of the parameters was increased in value. A much larger experiment was then carried out to provide a coarse-grained mapping of the parameter space, within the limits placed on the reaction parameters. The outcome of this experiment was also as expected from the chemistry of the reaction, but suggested that optimising this synthesis would be somewhat complicated. A large overlap was found between the mole fractions of the adducts; when there was a large amount of the most desirable adduct X_2 there was also a significant amount of unwanted X_3 , and when there was no X_3 the product was a mixture of X_1 and X_2 .

Based on these findings, multi-objective optimisations were carried out to balance the competing aims of maximising the amount of X_2 and minimising the amount of X_3 . This was done by using a weighted combination of $[X_2]$ and $[X_3]$ to reward the former and penalise the latter, and allowing both reagent flow rates and reaction temperature to vary. First, a working 2-dimensional self-optimisation was demonstrated using the SNOBFit algorithm, followed by a full 3-dimensional optimisation of the synthesis. While the bounds of the optimisation were changed to improve the volume of the parameter space being searched, the product composition found by the algorithm was unsatisfactory, as the weightings used in the objective function did not penalise production of X_3 enough. To confirm that an optimisation attempting to maximise $[X_2]$ would not provide satisfactory results either, another optimisation was carried out with $[X_2]$ as the sole objective. As expected, this yielded a product high in X_2 but also high in X_3 .

In summary, this chapter has described the development of an automated flow reactor for the synthesis and self-optimisation of *o*-xylenyl functionalised C_{60} derivatives. The reactor was shown to give stable, reproducible results, and as such provided a good platform for investigating and optimising the reaction. However, initial experiments mapping the parameter space suggested that optimising the reaction to give products of pure adducts would be an unrealistic aim. Similarly, multi-objective optimisations using a weighted combination of the mole fractions of the derivatives proved unsuccessful at gaining a product with the desired balance between the desirable and undesirable derivatives. The difficulty in choosing appropriate weights to penalise or reward production of the different derivatives in these multi-objective optimisations demonstrated that a more intuitive method was required to effectively optimise the trade-off between these products.

CHAPTER 4

Tuning Reaction Product
Composition by Constrained
Optimisation

4.1 Introduction

The previous chapter described the design and use of an automated flow system for the synthesis of *o*-xylenyl derivatised fullerene derivatives. This flow system was made of two syringe pumps that delivered the two reagents to a PTFE reaction channel, a PID-controlled solid-state heater, and an in-line HPLC system. All of these separate pieces of the flow system could be controlled by a computer, providing a stable and reproducible automated platform for synthesising fullerene derivatives. After performing an experiment to map the parameter space of the reaction by measuring the composition of the product synthesised at 246 different sets of conditions, it became clear that the synthesis presented a difficult challenge for optimisation due to the large overlap observed in the formation of the different derivatives.

As such, multi-objective self-optimisations were carried out that attempted to optimise the trade-off between maximising production of the doubly-substituted derivative (X_2) and suppressing the formation of the detrimental triply-substituted derivative (X_3), rather than solely maximising production of X_2 . A weighted sum of the mole fractions of X_2 and X_3 was used as the figure of merit for this trade-off. Although the optimisations worked well in the self-built automated flow reactor, they were unable to find a satisfactory product composition. Much of this problem was due to the difficulty of choosing suitable factors in the weighted sum figure of merit for rewarding and penalising production of the different derivatives.

Like the work described in the previous chapter, the examples of multi-objective self-optimisations of chemical syntheses in the literature use weighted-sum⁸⁹ and weighed-product^{88,90} based approaches to combined multiple targets into one figure of merit that an optimisation algorithm can use. These involve choosing appropriate weights to describe the relative importance, or desirability, of each target^{91–93}. However, as seen with our self-optimisations, it is often not obvious what weights should be used to achieve a particular objective. Often some experimentation, or mathematical knowledge, is needed before suitable weights and a working combined objective function are found. Therefore, the problem with using weighted-sum or weighted-product merit functions for multi-objective chemical optimisations is that there is often a disconnect between the knowledge about what a chemist might want from a chemical synthesis and converting that knowledge into some numerical form of weights for the individual merit functions.

The work in this chapter attempts to address the need for an easy to use and intuitive method for encoding the chemist’s knowledge about the requirements of a chemical synthesis by formulating the problem as a soft constrained optimisation. We explain how this approach has many advantages over conventional weighted sum and weighted product based merit functions.

4.2 Constrained Optimisation

Reformulating multi-objective optimisations as constrained optimisations may provide a more intuitive way of encoding a chemist’s requirements of a chemical synthesis into an optimisation than using a weighted combination of individual merit functions. In a constrained optimisation, as well as minimising the objective function, constraint functions are used to transform the parameter space into feasible and infeasible regions. Here we used a special kind of constrained optimisation known as a soft constrained optimisation. Unlike a normal constrained optimisation, the boundary between the feasible and infeasible regions is sloped, rather than a step function, allowing for semi-feasible points.

This type of optimisation is a good fit for the problem of the cascadic formation of fullerene derivatives by Diels-Alder cycloaddition. There is a large overlap in the distribution of each species, ruling out the selective synthesis of just one product. As such, there is a need to balance production of the desired X_2 with the formation of X_3 , which is detrimental to the application performance. A constraint function placed on the amount of X_3 can encapsulate this need, limiting it to below a particular value. However, it is unknown exactly how much X_3 is tolerable in the product, and it may be that a small violation of the constraint placed upon it is acceptable if it leads to an overall increase in the amount of X_2 . In this case, the amount of X_2 in the product would be used as the objective function, the amount of X_3 that is tolerable in the product would be used as the constraint function, and the amount of violation that is acceptable would be used as the softness of the constraint boundary.

The original paper by Huyer and Neumaier⁹⁵ describing SNOBFit, included a method for running soft constrained optimisations using the SNOBFit algorithm. A transformation of the objective and constraint functions, first proposed by Dallwig *et al*¹⁰⁸, was used to provide a single target for the optimisation. This target took the form of a soft merit function, $f(\mathbf{x})$ with two parts to it (Eqn. 4.1).

$$f(\mathbf{x}) = f^*(\mathbf{x}) + p(\mathbf{x}) \quad (4.1)$$

The first is a scaled objective function, $f^*(\mathbf{x})$, taking the form shown in Eqn. 4.2.

$$f^*(\mathbf{x}) = \frac{f(\mathbf{x}) - f_0}{\Delta + |f(\mathbf{x}) - f_0|} \quad (4.2)$$

where the value of the objective function, $f(\mathbf{x})$, is scaled by f_0 , the value of the objective at a previously identified feasible point $\mathbf{x}_0$, and Δ , a positive integer. The choice of Δ is flexible, but is usually taken as the median value of the absolute difference between f_0 and $f(\mathbf{x})$. Taking this form the scaled objective function $f^*(\mathbf{x})$ has a minimum of -1 and a maximum of 1.

The meaning of Δ is demonstrated in Fig. 4.1a, where $f^*(\mathbf{x})$ is plotted against $f(\mathbf{x})$

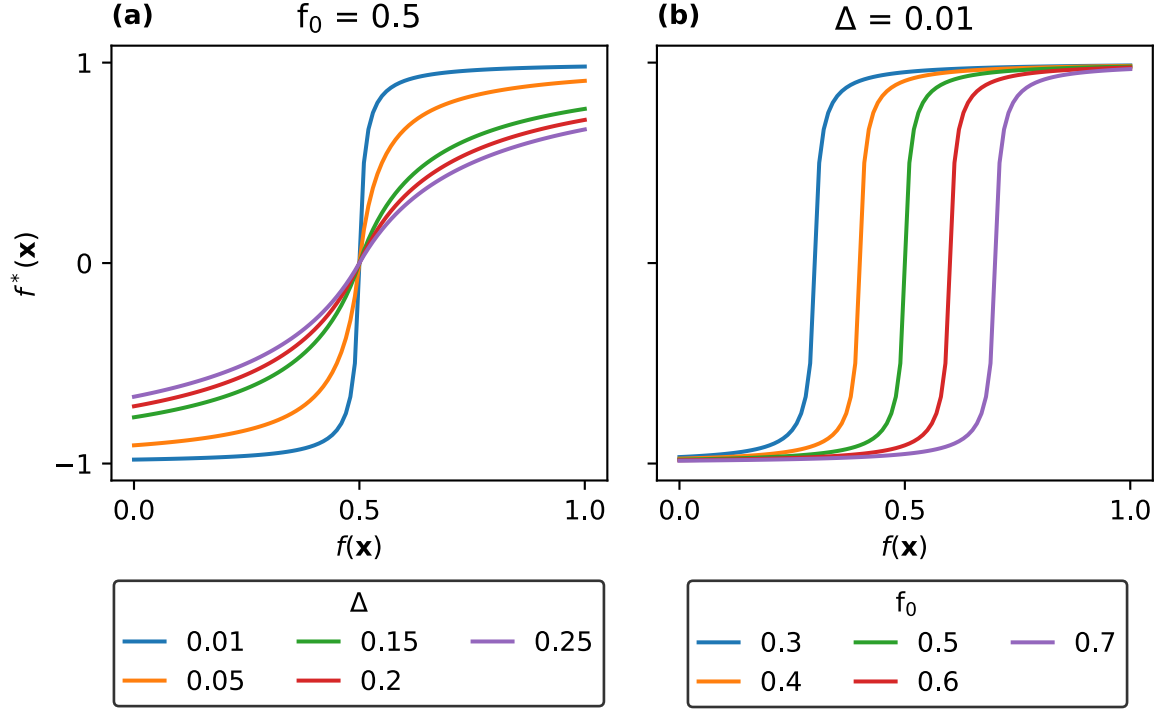


Figure 4.1: A demonstration of how the scaled objective function, $f^*(\mathbf{x})$, changes with the objective function, $f(\mathbf{x})$, when (a) Δ is varied from 0.01 to 0.25 while f_0 is held at 0.5 and when (b) f_0 is increased from 0.3 to 0.7 while Δ is held at 0.01.

for different values of Δ , while f_0 is held at 0.5. At the lowest value of Δ , 0.01, $f^*(\mathbf{x})$ resembles a step-function, rising sharply from the minimum of -1 to the maximum of 1 only when $f(\mathbf{x})$ is near to the chosen value of f_0 , 0.5. As Δ increases, the curve becomes shallower, but is still centred about the value of f_0 .

Setting Δ as the median difference between f_0 and some initial set of objective function values, sampled randomly from the parameter space, allows the true range of $f(\mathbf{x})$ to be mapped evenly to $f^*(\mathbf{x})$. However, if the initial feasible point found for f_0 is much higher than other subsequent feasible points, $f(\mathbf{x})$ could be unevenly mapped to $f^*(\mathbf{x})$ so that most feasible points are close to -1. The constrained optimisation files provided by Huyer and Neumaier attempt to counteract this by recalculating f_0 and Δ once, if any of the values of $f^*(\mathbf{x})$ are below than zero and a more optimum feasible point has been found.

The effect of changing f_0 on $f^*(\mathbf{x})$ is shown in Fig. 4.1b, with Δ held at 0.01. As f_0 is the centre point of the s-shaped curve, it has the expected behaviour of shifting the curve to the right as f_0 increases, or to the left as f_0 decreases. This represents moving the boundary where values of $f^*(\mathbf{x})$ transition from negative to positive.

The second part of the soft-merit function is the penalty function, $p(\mathbf{x})$, applied to the constraints (Eqn. 4.3).

$$p(\mathbf{x}) = 2 \frac{\sum_i \delta_i^2(\mathbf{x})}{1 + \sum_i \delta_i^2(\mathbf{x})} \quad (4.3)$$

Where $\delta_i(\mathbf{x})$ is a measure of how much point $\mathbf{x}$ violates the i -th constraint. The calculation for this violation is shown in Eqn. 4.4.

$$\delta_i(\mathbf{x}) = \begin{cases} F_i(\mathbf{x}) - F_i^{\text{lower}}/\sigma_i & \text{if } F_i(x) \leq F_i^{\text{lower}} \\ F_i(\mathbf{x}) - F_i^{\text{upper}}/\sigma_i & \text{if } F_i(x) \geq F_i^{\text{upper}} \\ 0 & \text{otherwise,} \end{cases} \quad (4.4)$$

This means that if the value of the constraint function, $F_i(\mathbf{x})$, is outside its limits, F_i^{lower} and F_i^{upper} , then $\delta_i(\mathbf{x})$ is equal to the amount by which it is outside these limits divided by a scaling factor, σ . If, however, $F_i(\mathbf{x})$ is within its limits, then $\delta_i(\mathbf{x})$ is zero. This in turn makes the overall penalty function zero - meaning that it is a feasible point. This penalty function has a maximum of 2 and a minimum of 0.

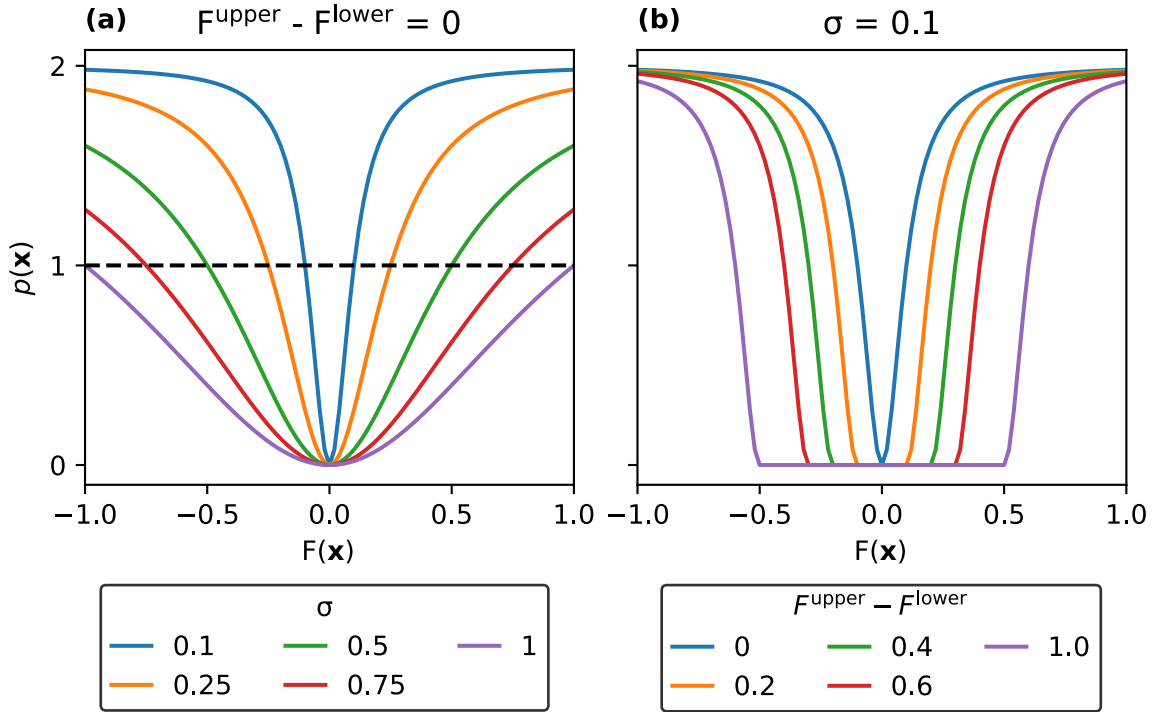


Figure 4.2: A demonstration of how the penalty function, $p(\mathbf{x})$, changes with the constraint function value, $F(\mathbf{x})$, when (a) σ is varied between 0.1 and 1 while $F^{\text{upper}} - F^{\text{lower}}$ is held at 0 and when (b) $F^{\text{upper}} - F^{\text{lower}}$ (the difference between the constraint limits) are varied between 0 and 1 while σ is held at 0.1.

The meaning of σ has been explored by plotting the value of $p(\mathbf{x})$ as σ is increased from 0.1 to 1 in Fig. 4.2a, with both the upper and lower limit of the single constraint function as zero. When σ is small, the value of the penalty function increases sharply as the constraint function moves above and below the limits placed on it, quickly reaching the maximum value of 2. As σ increases, the penalty function increases at a slower rate. In this way, σ can loosely be interpreted as the “steepness” of the penalty function close to the constraint boundaries - how fast it reaches its maximum. Another way of looking at

σ is as the maximum tolerated deviation outside of the limits of the constraint function. Looking at Fig. 4.2a, it can be seen that the penalty function has a value of 1 when the constraint function is outside of its limits by an amount equal to σ . As such, all points with a penalty function below 1 but outside the limits can be considered semi-feasible points, while all points above 1 are infeasible.

As would be expected, changing the limits placed on the constraint function extends the feasible region, where the penalty function is zero. In Fig. 4.2b, the penalty function has been plotted against the value of a hypothetical constraint function, with a constant σ of 0.01. The limits of the constraint function have been increased symmetrically about 0, from a width of 0.0 to a width of 1.0. The feasible region appears as a flattening of the bottom of the curve of the penalty function, growing as the limits are widened.

In the case of the synthesis of *o*-xylenyl derivatised fullerenes, there might be the need to apply multiple constraint functions to the optimisation. For example, a constraint may be placed upon the amount of X_3 in the product, as well as the total conversion from C_{60} . The effect that changing σ would have on a penalty function composed of two constraints is displayed in Fig. 4.3.

A separate σ is used for each constraint in the penalty function (Eqn. 4.3). The example in Fig. 4.3 uses two constraints, and so the penalty function has two values for σ - σ_1 for the first constraint $F_1(\mathbf{x})$ and σ_2 for the second constraint $F_2(\mathbf{x})$. To make it easier to see the effect that changing the two different σ values have on the penalty function, the penalty function value has been plotted against the violation of just the upper limits of the two constraints.

The value of σ_1 varies with each row of Fig. 4.3, from 0.1 at the top row to 1 at the bottom row, while σ_2 varies with each column, and from 0.1 in the left-most column to 1 in the right-most. A red line has been used to indicate the demarcation between the semi-feasible region, where the penalty function is below 1, and the infeasible region, where the penalty function is above 1. When σ_1 and σ_2 are the same, the penalty function is isotropic, increasing at the same rate as the violations of F_1^{upper} or F_2^{upper} increase. In the top right plot, both are at the lowest value of 0.1 and the semi-feasible region is very small. As σ_2 is increased, moving right across the plots, the shape of the semi-feasible region expands upwards to form an elongated channel at the left of the plots, where the violation of F_1^{upper} is low. The same elongation of the semi-feasible region is seen going down the left-most column of plots, where σ_1 is increased and σ_2 is held at 0.1, but along the bottom edge of the plots where the violation of F_2^{upper} is low.

Given the form of the penalty function, it behaves exactly as one would expect when σ and the constraint limits are changed. A more interesting picture, and one that is more relevant to the optimisation in hand, is gained by plotting what the soft merit function would look like with real data from the synthesis of fullerene derivatives, to capture the trade-off between the yields of the different derivatives. This was done by taking the

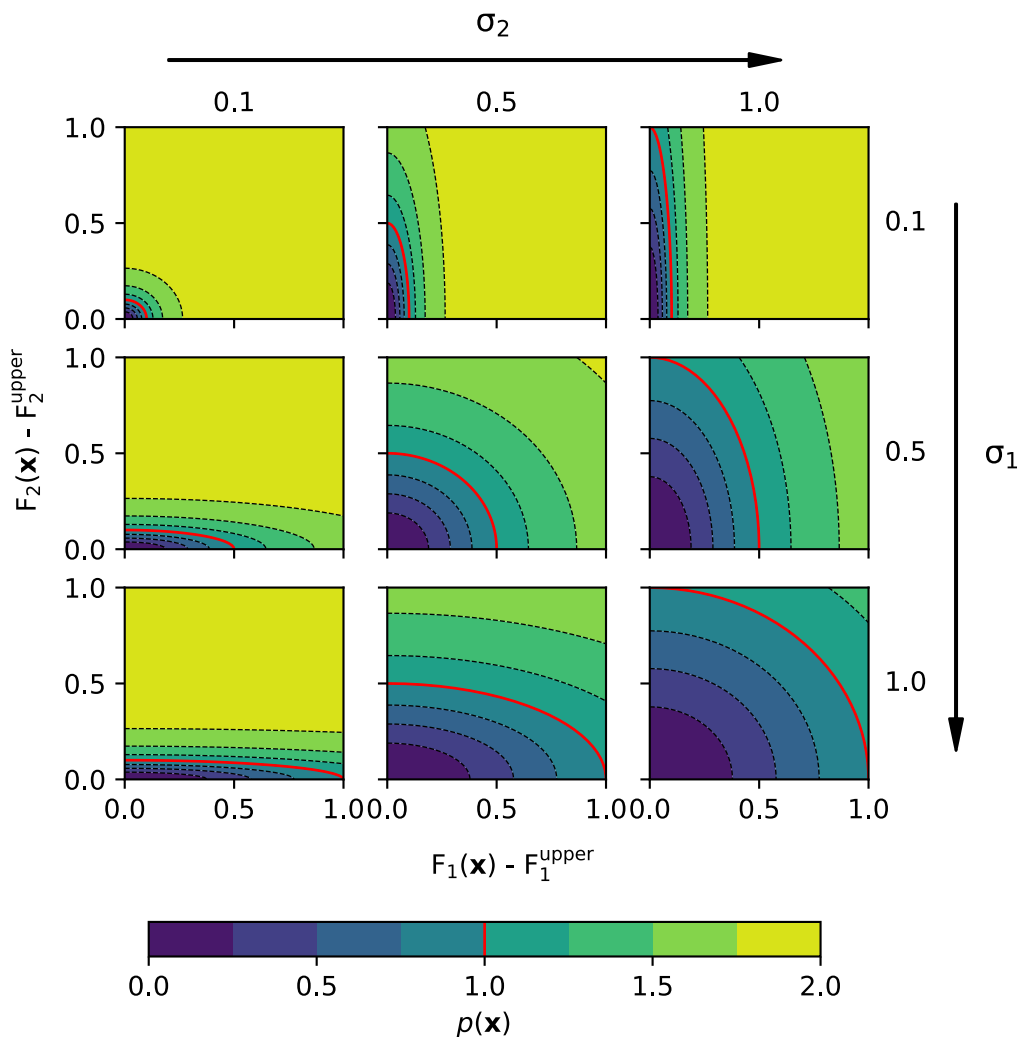


Figure 4.3: Demonstration of the effect that the interaction between two constraint functions, $F_1(\mathbf{x})$ and $F_2(\mathbf{x})$, has on the penalty function, $p(\mathbf{x})$, as the two different “softness” parameters, σ_1 and σ_2 , are changed from 0.1 to 1.

measured product composition from the systematic search of the parameter space carried out in the previous chapter. Points were selected so that the effect that increasing the amount of X_2 in the product has on the soft merit function could be visualised. This was done by finding three product compositions with a mole fraction of X_2 closest to the values in the series between 0 and 0.5, at intervals of 0.05, and choosing the composition with the lowest fraction of X_3 .

As the most useful product of the reaction is X_2 , the objective function used to calculate the soft merit value was chosen to maximise its mole fraction (Eqn. 4.5).

$$f(\mathbf{x}) = 1 - [X_2] \quad (4.5)$$

The least desired, and actively detrimental product, is X_3 . Therefore, a single constraint function was used to limit the amount of X_3 in the product to below a chosen mole

fraction, F^{upper} . This is shown in Eqn. 4.6.

$$F(\mathbf{x}) = [X_2] < F^{\text{upper}} \quad (4.6)$$

The soft merit function is shown versus the objective function in Fig. 4.4a-d, as each of the parameters of the scaling and penalty functions discussed above are changed.

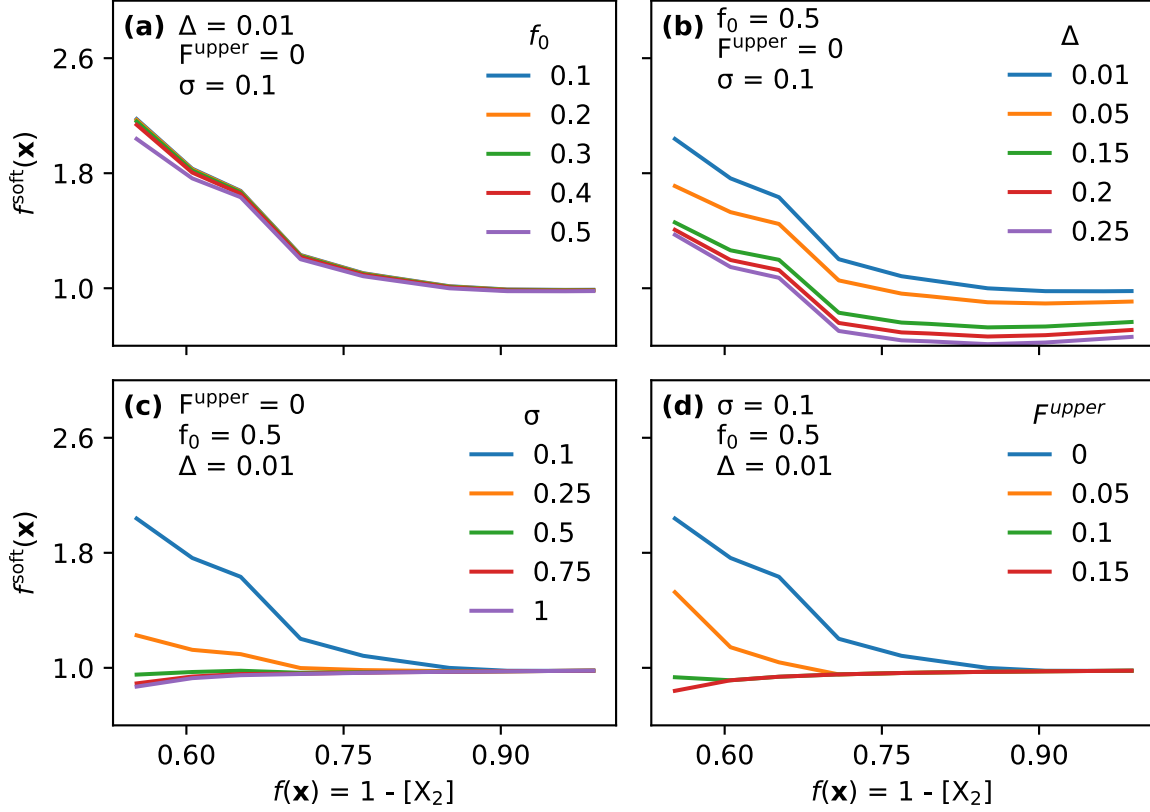


Figure 4.4: Comparison of how each of the defining parameters, (a) f_0 , (b) Δ , (c) σ , and (d) the constraint limit F^{upper} , affect the soft merit function, $f^{\text{soft}}(\mathbf{x})$, when an example set of objective and constraint functions, Eqns. 4.5 and 4.6, are calculated using real data taken from the parameter space mapping experiment from the previous chapter.

Little change was seen when f_0 was increased from 0.1 to 0.5, and all other parameters held constant (Fig. 4.4a). Changing Δ from 0.01 to 0.25 (Fig. 4.4b) had a bigger effect, and moved the soft merit value curve down as Δ was increased. This suggests that in isolation, the choice of f_0 has little effect on the soft merit function, and therefore the optimisation. However, Δ is usually chosen as the median difference between f_0 and the objective function value for some set of starting points, and so in this context f_0 will affect the optimisation. Lowering the values of the soft merit function should not have a significant effect on finding an optimum point in this case, as the shape and curvature of the function would not have changed, but could prove detrimental if the values are close to the minimum of the soft merit function, -1. In that case, a Δ that is too low could artificially flatten out the soft merit function around its minimum, possibly missing the

true global minimum.

The parameters of the penalty function displayed more influence on the soft merit value. As would be expected, once $[X_2]$ increased above a certain point, $[X_3]$ started to increase as well, and so the soft merit function increased as the objective function decreased. This was seen clearly when σ_1 was 0.1, in Fig. 4.4a. As σ_1 increased, the increase in the soft merit function became slower, until $f^{\text{soft}}(\mathbf{x})$ actually decreased with increasing $f(\mathbf{x})$, at $\sigma = 0.75$ and 1. A similar picture was seen when changing the upper limit of the constraint function (Fig. 4.4b) where the start of the increase in $f^{\text{soft}}(\mathbf{x})$ was pushed from objective function values of 0.9, to 0.75, and then below 0.6. From this it is clear that choosing the right value for σ and the right limits on the constraint function are very important. Most importantly, if the boundary of the constraint function is too soft, i.e. σ is too high, then the penalty function may have little to no effect on the soft merit function.

While the choice of the limits on the constraint function may be clear from the desired product specification, or other outside considerations, the choice of σ is a little harder. On the one hand it may be thought of as the maximum tolerated violation of the constraints, marking the transition from feasible and semi-feasible points to infeasible points. However, it can also effect how long the optimisation algorithm takes to find a feasible point, with smaller values allowing the algorithm to find feasible points more accurately, but may mean that it takes a longer time to find them, or does not find them at all.

4.2.1 Changing sigma

Ideally we would like to maximise the amount of X_2 while completely suppressing formation of X_3 . The work described in the previous chapter showed how difficult this was to achieve when just maximising the amount of X_2 , as well when using a weighted sum based merit function that rewarded production of X_2 and penalised production of X_3 . It was hoped that using the constrained optimisation approach would make this goal more achievable, but we will see from the results in this section that this is still an incredibly hard thing to do.

To formulate this problem as a constrained optimisation, the objective function (Eqn. 4.7a) was chosen to maximise the amount of X_2 , and a constraint was placed on the amount of X_3 in the product (Eqn. 4.7b). Both the upper and lower limits of the constraint were chosen as 0, to reflect that we wanted to completely suppress formation of X_3

$$f(\mathbf{x}) = 1 - [X_2] \quad (4.7a)$$

$$F(\mathbf{x}) = 0 < [X_3] < 0 \quad (4.7b)$$

The discussion in the previous section highlights how important the value of σ is to the outcome of these constrained optimisations. Changing the value of σ will change what we define as an acceptable violation of our constraint, and therefore how much X_3 we will tolerate in our product. In order to see how this choice would affect the best product composition found by our constrained optimisation, three optimisations were carried out with different σ values. The values were chosen as 1.5, 0.5, and 0.1. The choice of the highest value of σ as 1.5 might seem unusual for this system: the values that we are measuring are mole fractions, which can only take values between 0 and 1. This means that for the optimisation where σ was 1.5, all violations of the constraint were semi-feasible, and none could be fully infeasible. This was done to demonstrate the extreme where the penalty for violating the constraint was low for all violations.

All three optimisations were carried out within the same bounds, with overall flow rates between 30 and 310 μLmin^{-1} , flow rate ratios between 2 and 0.5, and temperatures between 100 and 150 $^{\circ}\text{C}$; residence times were between 3.0 and 30.7 min, and molar sulfate excesses between 1.5 and 6. Each was run for 140 measurements in total, or 19 calls to SNOBFit. Rather than using a set of random points to initialise the optimisations, and calculate f_0 , an unconstrained SNOBFit optimisation was performed to minimise the penalty function and search for an initial feasible point. This was done because it was thought that a feasible point would be hard to find. After finding a feasible point, or after evaluating the objective function 50 times, the routine would switch to the constrained optimisation - calculating the soft merit value for all the points investigated during initialisation and using them to inform the constrained optimisation.

The results for the soft optimisation can be seen in Fig. 4.5a when σ is 0.1, Fig. 4.5b when it is 0.5, and Fig. 4.5c when it is 1.5. Points in these scatter plots that have a lower $f^{\text{soft}}(\mathbf{x})$ value appear as dark purple circles, while points with a higher $f^{\text{soft}}(\mathbf{x})$ value appear as light yellow circles.

The results from these optimisations indicate that the optimisation run with $\sigma = 0.1$ had overall higher $f^{\text{soft}}(\mathbf{x})$ values than the optimisations using $\sigma = 0.5$ and $\sigma = 1.5$. There are more points that are lighter (i.e. have a higher $f^{\text{soft}}(\mathbf{x})$) in the scatter plot for $\sigma = 0.1$ (Fig. 4.5a) than there are in the scatter plots for $\sigma = 0.5$ (Fig. 4.5b) and $\sigma = 1.5$ (Fig. 4.5c). Similarly, the points in Fig. 4.5a with the lowest $f^{\text{soft}}(\mathbf{x})$ value are clustered towards the centre of the parameter space, while those in Fig. 4.5b cluster more towards the thin “spine” at the front of the parameter space, and the lowest value points in Fig. 4.5c tend to be located more towards the top corner of this “spine”. This suggests that as the value of σ used for the constrained optimisations was increased, the minimum of $f^{\text{soft}}(\mathbf{x})$ moved from the centre of the parameter space, where faster residence times suppress formation of X_3 , to longer residence times and higher temperatures that favour high production of both X_2 and X_3 . By changing σ from 0.1 to 1.5, the constraint on production of X_3 was significantly softened, allowing the optimisation to tolerate higher

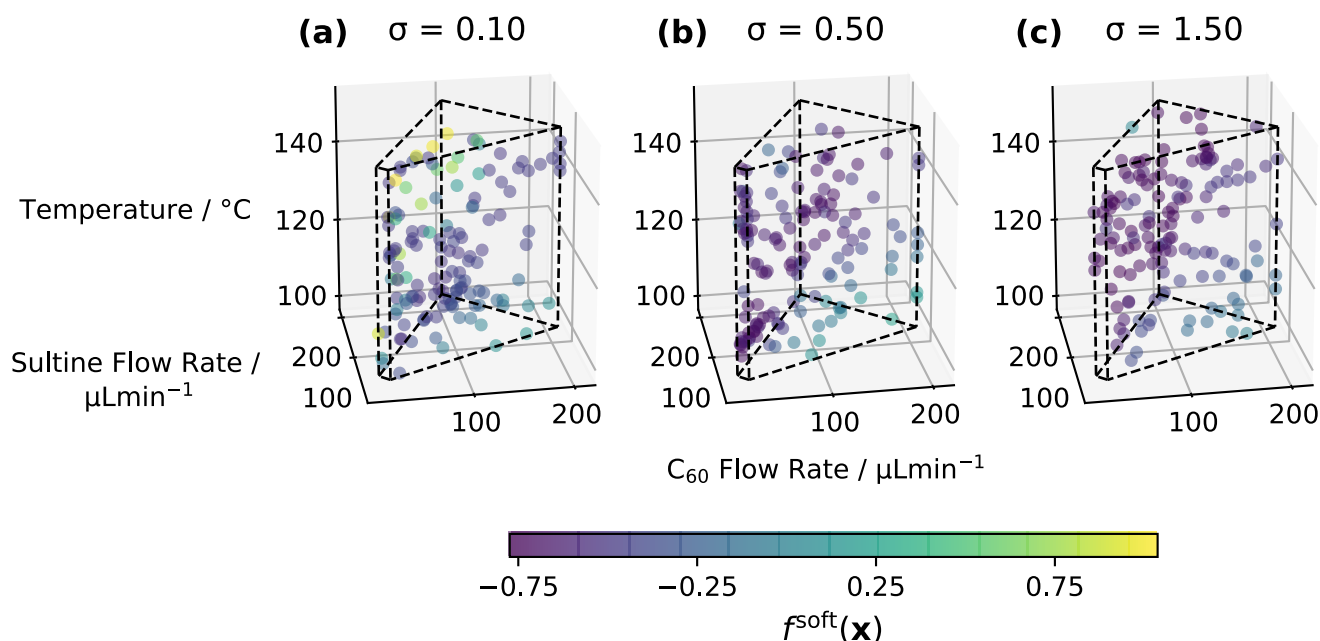


Figure 4.5: Comparison of the points sampled during three soft constrained SNOBFit optimisations with different values of σ : (a) 0.1, (b) 0.5, and (c) 1.5, shown as scatter plots with the colour of the points representing the value of the soft merit value, $f^{\text{soft}}(\mathbf{x})$. The same bounds were used for all three: 30 and 310 μLmin^{-1} for the overall flow rate, 2 and 0.5 for the ratio between the flow rates, and 150 °C and 100 °C for temperature. All optimisations were run for 140 measurements.

levels of X_3 if it meant increasing the amount of X_2 , and therefore moving the location of the minimum to reaction conditions that promote high levels of both X_2 and X_3 .

This is confirmed by looking at the mole fraction at the best point found after each call to the SNOBFit algorithm. When a σ value of 0.1 was used (Fig. 4.6a), the constraint boundary was relatively hard, which is reflected by the small area of X_3 at all best points. However, the amount of X_2 in the product was also kept low and X_1 was dominant product. The optimisation lowered the soft merit value from -0.477 at measurement 5, to -0.530 then -0.534, at measurements 10 and 41. The mole fraction of X_3 at these points was 0.011, 0.017, and 0.021, while the mole fraction of X_2 was 0.145, 0.191, and 0.211, and the amount of X_1 was 0.773, 0.746, and 0.734. From this it is clear that the amount of X_3 in the running best points was kept very low throughout. The optimisation successively improved the soft merit value by increasing the amount of X_2 in the product, and allowing the amount of X_3 to increase by only a small amount.

Increasing the value of σ to 0.5 (Fig. 4.6b) caused the composition of the product at the best points in the optimisation to have more X_2 and less X_1 , but also more X_3 . The best point remained the same for most of the optimisation (measurement 7, with a soft merit value of -0.720) but decreased once, at measurement 116, to give an overall best soft merit value of -0.724. In terms of the composition of the product, X_1 was still the major constituent with a mole fraction of 0.535 then 0.559. The amount of X_2 in the product

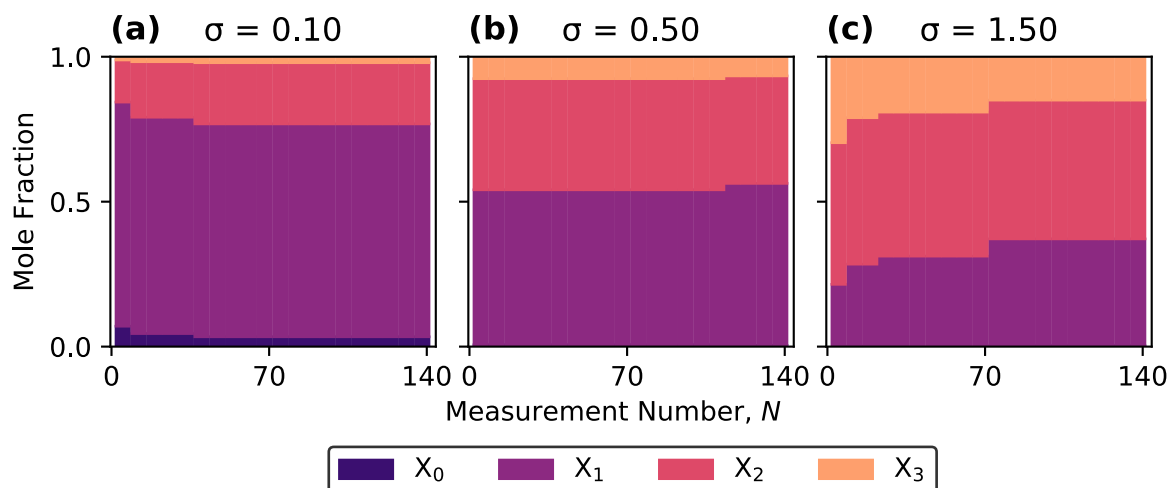


Figure 4.6: The composition of the best product mixture identified from each batch of points during soft constrained SNOBFit optimisations with $\sigma = 0.1$ (a), 0.5 (b), and 1.5 , (c).

was much better than when the optimisation was run with a σ of 0.1 , starting at 0.383 and reducing slightly to 0.371 , while the fraction of X_3 was higher - 0.076 at measurement 7 and 0.066 at measurement 116 .

The amount of X_2 and X_3 in the product increased further when the optimisation was run with a σ value of 1.5 , as seen in Fig. 4.6c. The soft merit value started at -0.722 at measurement 8 , and decreased to -0.763 then to -0.768 at measurements 9 and 28 , before reaching the overall lowest value of -0.774 at measurement 76 . The mole fraction of X_2 at these points was 0.489 , 0.505 , 0.497 , and 0.479 . For X_3 it was 0.297 , 0.210 , 0.191 , and 0.150 , and for X_1 it was 0.215 , 0.283 , 0.312 , and 0.369 . This clearly displays the same, but more exaggerated, trend as when σ was 0.5 .

Furthermore, the progression of the optimisation when σ was 1.5 and the composition of the optimum point found were very similar to the previous 3D optimisations that minimised Eqn. 3.1. As the constraint function of Eqn. 4.7b has a maximum value of 1 , using such a high value for σ meant that all of the points were semi-feasible. Given such a shallow gradient for the penalty function, all parts of the optimisation operate with a similar weighting on the penalty function, analogous the fixed penalty on X_3 demonstrated in the compound objective function.

This clearly shows how choice of σ in a soft optimisation can have profound effects on the location of the point with the lowest soft merit value. However, as can be seen from the compositions of the best points for each optimisation, none of the best points of the optimisations were feasible points - all had some X_3 in them, so violated the upper limit of the constraint function. This is an important point for these soft optimisations, what should be taken as the optimum point: the point with the overall lowest soft merit value, or the feasible point with the lowest soft merit value?

Chromatograms for the best and best feasible points for each optimisation are shown

in Figs. 4.7a(i)-(iii), with corresponding bar charts of their composition alongside in Figs. 4.7b(i)-(iii). Best feasible points were only found for the optimisations using σ of 0.1 and 1.5. From this plot, a few trends can be observed. As σ is decreased from 1.5 to 0.1, the amount of X_3 in the product decreased, while the amount of X_2 increased at first before decreasing again, and the amount of both X_1 and X_0 increased.

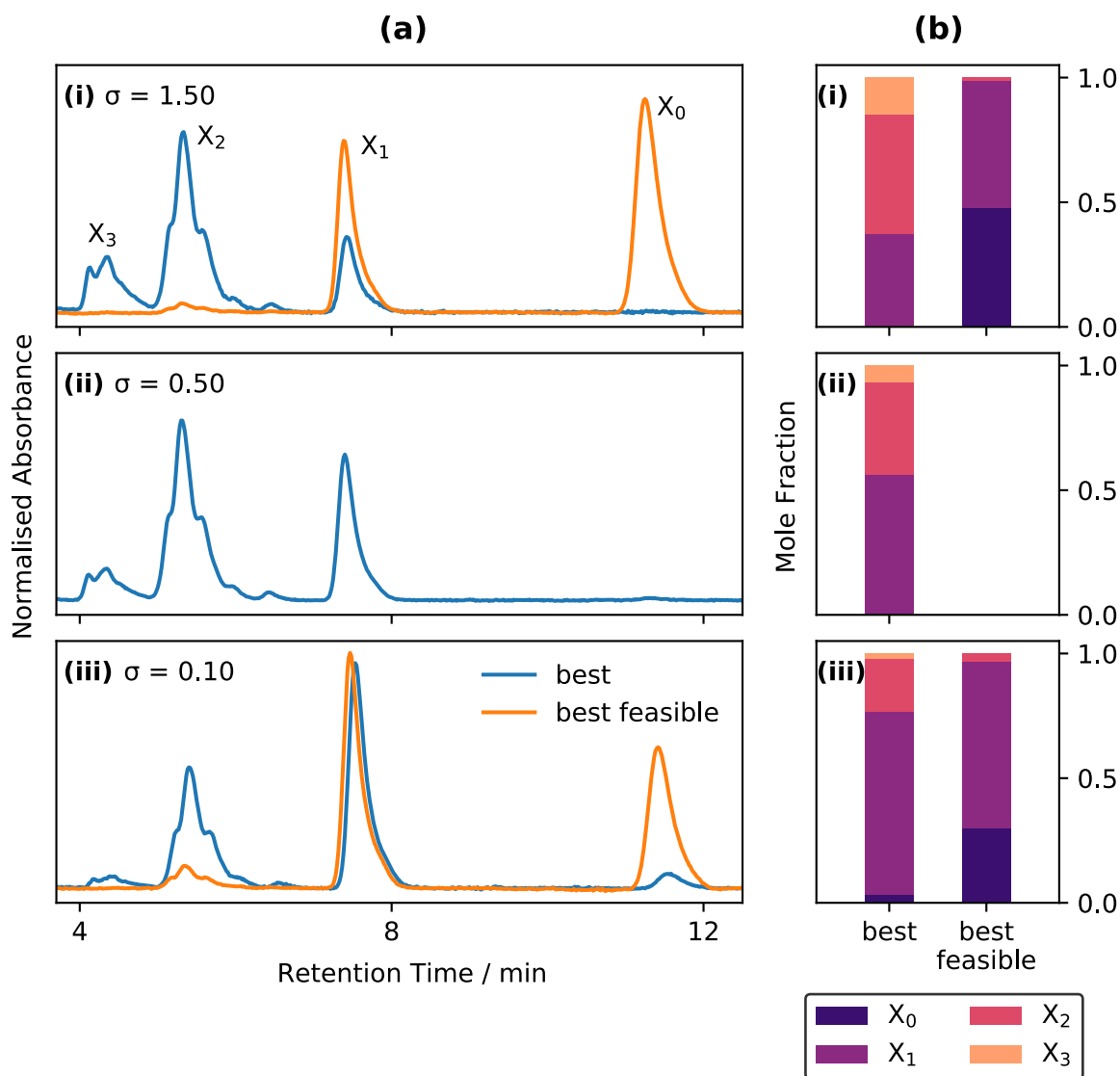


Figure 4.7: Chromatograms of the best and best feasible results identified during soft constrained optimisations when σ was 1.5 (a(i)), 0.5 (a(ii)), and 0.1 (a(iii)), alongside corresponding bar charts showing the composition at these points (b(i-iii)).

Given that best feasible points were only found for two optimisations, little can be said about any trend in the composition of these points. However, the best feasible point found for $\sigma = 1.5$ had a composition of 0.479, 0.509, 0.013, and 0.000 for X_0 , X_1 , X_2 , and X_3 , while the best feasible point found for $\sigma = 0.1$ had a composition of 0.301, 0.666, 0.033, and 0.000. It is notable that in satisfying the constraint and completely suppressing formation of X_3 , the best feasible points have very low amounts of X_2 . The stark difference

between the chromatograms for the best feasible points and those for the points with the lowest soft merit value is further evidence that our chosen objective of large amounts of X_2 with no X_3 was unachievable.

In this case, where the constraint used can not be satisfied or can only be satisfied at the cost of a poorly optimised objective function, it appears that it is best to take the point with the lowest soft merit value as the overall optimum, rather than the best feasible point. When making this choice, the interpretation of σ as the boundary between semi-feasible and infeasible points, and therefore the maximum tolerable violation of the constraints, becomes important.

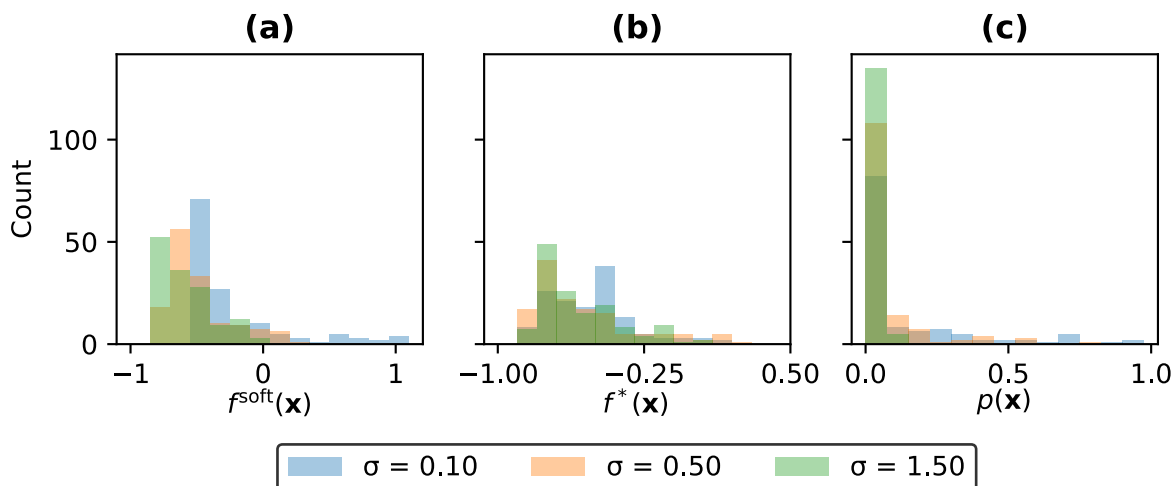


Figure 4.8: Distributions of values sampled for $f^{\text{soft}}(\mathbf{x})$ (a), $f^*(\mathbf{x})$ (b), and $p(\mathbf{x})$ (c), for soft constrained SNOBFit optimisations using different values for σ .

The importance of the gradient of the penalty function can be seen by looking at the distribution of values for the soft merit function, as well as the scaling and penalty functions, as σ was changed. The value of σ clearly had a significant effect on the distribution of values of the soft merit function, shown in Fig. 4.8a. In all of the optimisations, the distribution was skewed towards lower values. However, the maximum of the distribution shifted to lower values as σ was increased. The distribution of values for the scaling function, in Fig. 4.8b, was largely unchanged when varying σ , although the distribution for $\sigma = 0.1$ was slightly broader than the other two. The change in distribution of the penalty function (Fig. 4.8c) was more significant. For all three optimisations, the maximum of the distribution was near 0, but this maximum grew as σ increased - as would be expected if the penalty for violating the bounds of the optimisation was reduced. Despite being the obvious result of reducing the maximum penalty over the range of the mole fractions, this also suggests that the lower the value of σ used in an optimisation, the more weighted the soft merit function is towards penalising violation of the constraints than it is to minimising the objective function.

In Fig. 4.9a, the soft merit function has been plotted against the value of the constraint

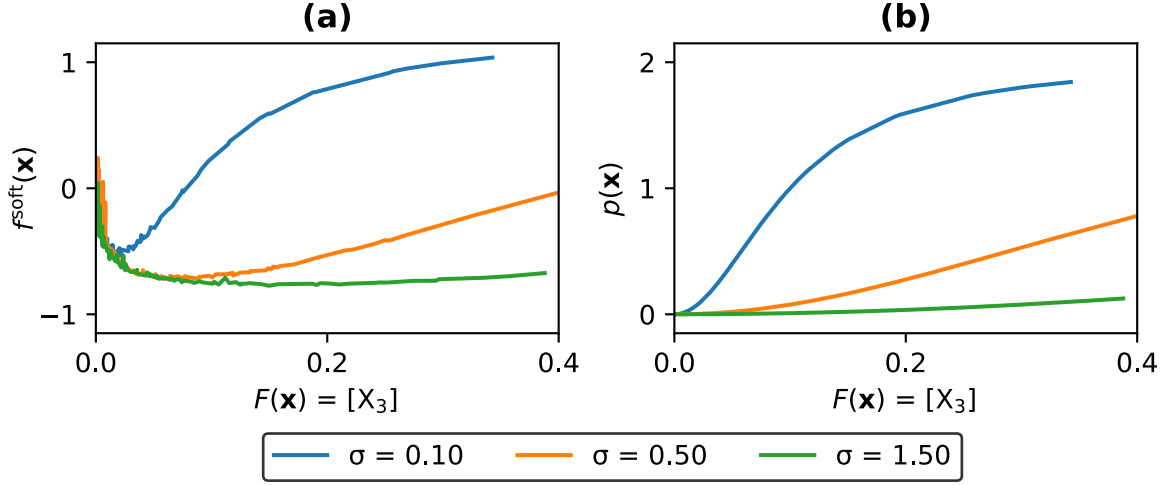


Figure 4.9: Comparison of how $f^{\text{soft}}(\mathbf{x})$ (a) and $p(\mathbf{x})$ (b) change with $[X_3]$, the constraint function in Eqn. 4.7b, for soft constrained SNOBFit optimisations using different values for σ .

function, $[X_3]$. For all values of σ , there is a region near zero where the soft merit value takes a small range of values, as this is either in or near the feasible region, so the soft merit function does not depend strongly on the penalty function. As $[X_3]$ starts to increase, $f(\mathbf{x})$ decreases, due to an associated increase in $[X_2]$. For the optimisation when σ was 0.1, the soft merit function quickly reaches a minimum, then increases rapidly with increasing $[X_3]$. Both of the other optimisations also show $f(\mathbf{x})$ decreasing to a minimum, before eventually rising again. However, the increase seen when σ was 0.5 was much shallower, and the shape of the curve when σ was 1.5 is almost flat, before increasing slightly at the highest amounts of X_3 seen.

Looking at Fig. 4.9b, where $p(\mathbf{x})$ is shown against $[X_3]$, it can be seen that the penalty function changes with the constraint function as expected from the plots above. For $\sigma = 0.1$, the function increases rapidly between $[X_3] = 0$ and $[X_3] = 0.1$. The curve then starts to even-off as it increases towards its maximum value of 2. As $[X_3]$ never approaches the value of σ for $\sigma = 0.5$, the curve of the penalty plot for this optimisation never starts to even off, instead increasing with a shallower gradient than for $\sigma = 0.1$. Finally, the curve for $\sigma = 1.5$ only starts to increase with higher values of $[X_3]$, and only by a small amount.

Comparing Fig. 4.9a to Fig. 4.9b, it appears that when the constraint function value is low, $f(\mathbf{x})$ can be a range of values as it depends most strongly on the scaled objective function, $f^*(\mathbf{x})$. As the constraint function increases, after passing through a minimum value, the shape of $f(\mathbf{x})$ begins to resemble the shape of the penalty function for all three of the optimisations.

However, the difference between these real optimisations and the example functions used in the previous section, is that the objective and the constraint functions are linked. As the objective function, Eqn. 4.7a, decreases, the constraint function, Eqn. 4.7b, increases, but at a slower rate. The effect that this has on the soft merit function, $f(\mathbf{x})$, can

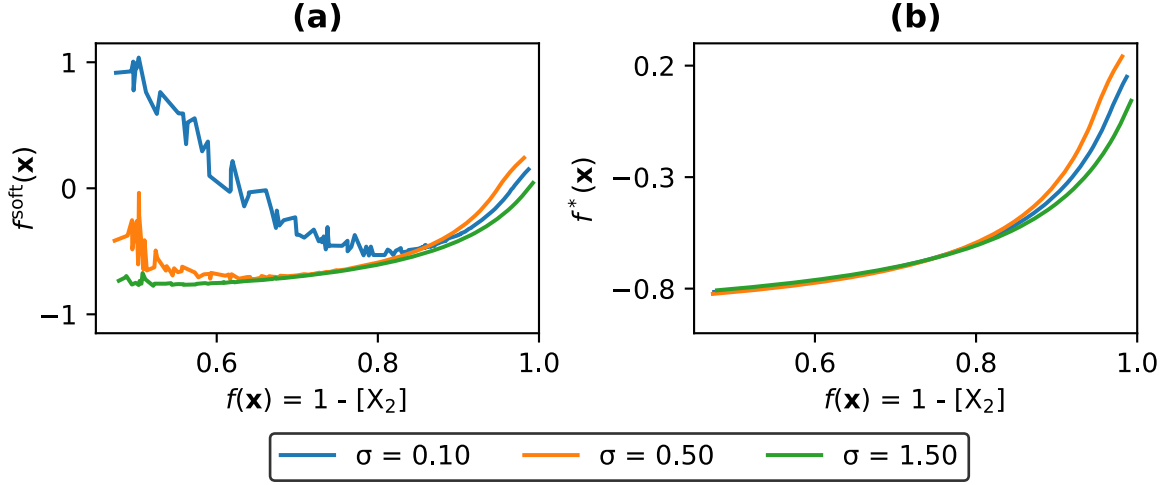


Figure 4.10: Comparison of how $f^{\text{soft}}(\mathbf{x})$ (a) and $f^*(\mathbf{x})$ (b) change with $1 - [X_2]$, the objective function in Eqn. 4.7a, for soft constrained SNOBFit optimisations using different values for σ .

be seen in Fig. 4.10, where the soft merit value at the corresponding objective function values is shown.

For all three optimisations, when $f(\mathbf{x})$ is at its highest, $f^{\text{soft}}(\mathbf{x})$ is also high. As would be expected, a decrease in the objective function is reflected in a decrease in $f^{\text{soft}}(\mathbf{x})$. However, when σ is 0.1, decreasing the objective function past 0.8 leads to a relatively sharp increase in $p(\mathbf{x})$ and therefore $f^{\text{soft}}(\mathbf{x})$. For $\sigma = 0.5$, $f^{\text{soft}}(\mathbf{x})$ continues to decrease with decreasing $f(\mathbf{x})$, but again increases when it gets below 0.5. Finally, when σ is 1.5, $f^{\text{soft}}(\mathbf{x})$ continues decreasing with $f(\mathbf{x})$, and does not see the increase that the other two saw. This demonstrates the link between the constraint and objective functions: once the objective function decreased below a particular value, the contribution of the penalty function to the soft merit function began to increase. This again demonstrates that as σ is increased, the influence of the penalty function on the soft merit function decreases.

To confirm this, the scaled objective function, $f^*(\mathbf{x})$, has been plotted against the objective function. All three of the curves behave as expected, decreasing with decreasing objective function. That there is little difference between them suggests that there is no dependence on σ . However, there is a difference at higher values of the objective function, with the curve for σ of 0.5 rising the highest, followed by 0.1, then 0.1. The only difference between these curves would be because of the values for f_0 and Δ used in each of the optimisations. The f_0 value for each of the optimisations was 0.987, 0.949, and 0.967 for a σ of 0.1, 0.5, and 1.5 respectively. For Δ , the values were 0.121, 0.102, and 0.111. These differences in the scaling parameters are enough to make a difference only at higher values of the objective function.

The effect that this has on the soft merit value can be better understood by looking at how the soft merit function, the objective function, and the constraint function all vary as subsequent best points are found in each of the optimisations. Unsurprisingly, as the

soft merit value is the quantity being optimised by these constrained optimisations, it decreases with each new best point, shown in Fig. 4.11a. As seen above, the soft merit values were highest for $\sigma = 0.1$, followed by $\sigma = 0.5$, and then $\sigma = 1.5$.

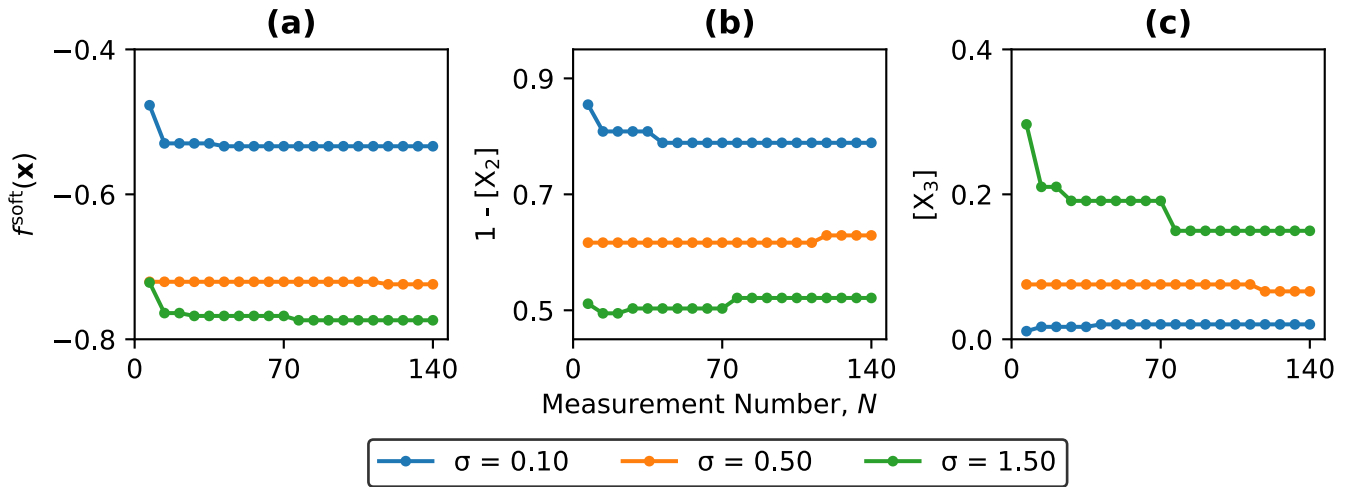


Figure 4.11: Illustration of how $f^{\text{soft}}(\mathbf{x})$ (a), the objective function $1 - [X_2]$ (b), and the constraint function $[X_3]$ (c) at the current best result change over the course of the soft constrained SNOBfit optimisations using different values for σ .

From Fig. 4.11b, it can be seen that for $\sigma = 0.1$ the reduction in the soft merit function was due to a reduction in the objective function. However, for $\sigma = 0.5$, the objective function remained the same for most of the optimisation, before increasing slightly at the end, while for $\sigma = 1.5$ the objective function decreased at first before increasing again.

This is in contrast with the behaviour of the constraint function, shown in Fig. 4.11c. It is clear that by having the largest σ value, and therefore lowest penalty for constraint violations, $\sigma = 1.5$ had the largest constraint function values throughout, as well as the largest reduction in the constraint function with each successive best point. This compensated for the moderate increase in objective function at the end. The same decrease in the constraint function was seen for $\sigma = 0.5$, but for $\sigma = 0.1$ the constraint function increases slightly with each new best point found.

These results show that the choice of σ has important ramifications for how a soft constrained SNOBfit optimisation proceeds. Choosing too small a value leads to a severe penalty function that is large even for small violations. A large penalty like this can swamp the objective function, and lead to an optimum that satisfies or is close to satisfying the constraints but has a poorly minimised objective. In the case of the experiments described in this section, this highlights just how unachievable our original objective was. Changing the value of σ from very low to very high changed the product composition from suppressing formation of X_3 but with very little X_2 to producing large amounts of X_2 but with unacceptable levels of X_3 .

4.2.2 Selecting Product Composition

The reason for using constrained optimisations was to provide an easier, more intuitive way of incorporating a chemist’s requirements for a chemical synthesis into a self-optimisation, exemplified by the cascadic synthesis of fullerene derivatives. So far, the use of the SNOB-Fit algorithm with an automated flow reactor has been demonstrated for this synthesis, and in the previous section a merit function that combined optimising a scaled objective function with a penalty applied to constraint functions was explored. From this it was seen that in terms of the synthesis of *o*-xylenyl functionalised fullerene derivatives, the best way to formulate the specification of this reaction was to frame it as a minimisation of the amount of X_3 in the product, with constraints placed on the other products.

In this section, an attempt is made to demonstrate how a chemist’s knowledge of the requirements of a chemical synthesis, in the form of a product specification, can be encoded into a chemical optimisation using the soft merit function previously explored. As mentioned previously, the triply-substituted product, X_3 , is the most undesirable product. Therefore, the best outcome is to have as little X_3 as possible, subject to whatever other constraints are placed on the product composition. Previous optimisations have attempted to maximise the amount of X_2 in the product, as it is the most useful for OPV applications. However, the results have shown that it is not possible to have large amounts of X_2 in the product without the presence of significant amounts of either X_1 or X_3 . While X_3 is very detrimental to application performance, as previously said, X_1 can actually be used for some applications, as well as being able to be recycled as a feedstock for producing X_2 . For these reasons, we decided to minimise the amount of X_3 , while ensuring a product that was composed of over 90% X_2 and X_1 .

The formulation of this into a soft constrained optimisation was done using the objective function Eqn. 4.8a and the constraint function Eqn. 4.8b.

$$f(\mathbf{x}) = [X_3] \quad (4.8a)$$

$$F(\mathbf{x}) = 0.9 < [X_2] + [X_1] \quad (4.8b)$$

Translating the need for as little X_3 as possible while keeping the amount of X_1 and X_2 above a threshold value using this method is clearly much easier than selecting the right set of weights to do the equivalent with an optimisation that uses a weighted sum or product based merit function.

In order to demonstrate the utility of using a soft constrained optimisation to encode a product specification, two optimisations were carried out. The first, Run IVa, was an unconstrained SNOBFit optimisation using the objective function $f(\mathbf{x})$ above. The second, Run IVb, was a soft constrained optimisation using both $f(\mathbf{x})$ and the constraint

function $F(\mathbf{x})$. For Run IVb, σ was set to 0.3. This could be interpreted as specifying that a product composition with anything down to 60% X_1 and X_2 would be acceptable. However, the main reason for choosing this value was to avoid the pitfalls of using a too stringent (i.e. small) σ value, as discussed above. Both optimisations were run for 100 measurements, and the points sampled during each can be seen in Fig. 4.12a and b.

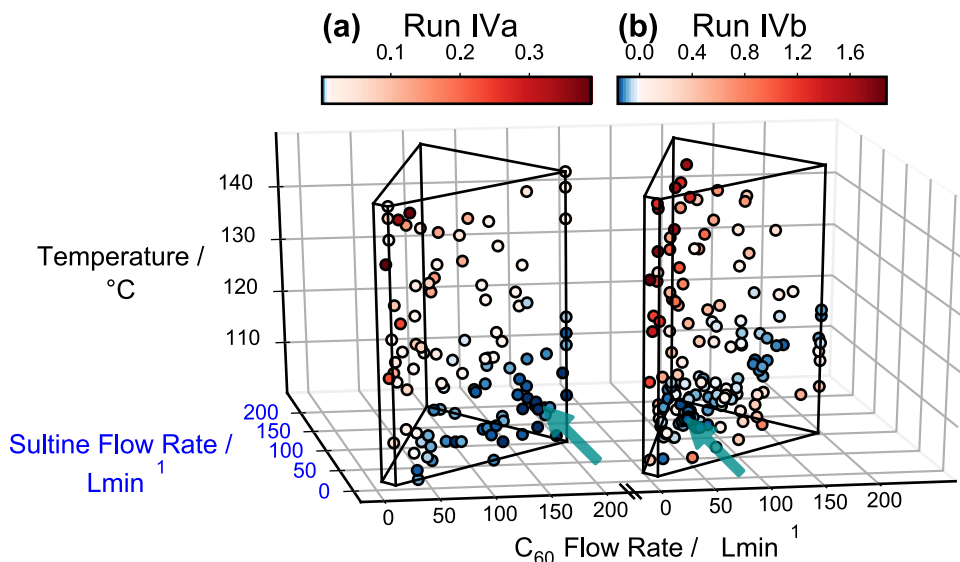


Figure 4.12: Comparison of the points sampled from a SNOBFit optimisation, Run IVa (a), and a soft SNOBFit optimisation, Run IVb (b). The colour of each point represents the difference between the merit value at that point and the median merit value for the optimisation, 0.011 for Run IVa and 0.020 for Run IVb. The objective function of both runs and the constraint function of RunIVb, are defined in Eqns. 4.8a and Eqns. 4.8b. The same bounds were used for both runs: 310 and 30 μLmin^{-1} for the overall flow rate, 2 and 0.5 for the ratio between the flow rates, and 150 $^{\circ}\text{C}$ and 100 $^{\circ}\text{C}$ for temperature. Both were run for 100 measurements.

The points sampled during Run IVa are shown on the left side of the figure. The colours of the markers denote their objective function value relative to the median value of 0.011, with stronger more saturated colours representing values further from the median. For ease of distinction, values below the median have been coloured blue, and values above the median red, while those close to or at the median are white. Preferential sampling by the SNOBFit algorithm can be seen by the large cluster of blue to dark blue markers towards the bottom right corner of the right face of the bounding cage. The dark blue of these markers show that they have function values furthest below the median. Indeed, the best point found during the optimisation is near the centre of them, denoted by a large blue arrow. At this minimum the merit value was 0.001, with the mole fractions of X_0 , X_1 , X_2 , and X_3 being 0.318, 0.649, 0.033, and 0.001. On the right of Fig. 4.12 is a scatter plot of the points sampled during the soft constrained optimisation of Run IVb. The markers have been coloured according to their soft merit values, relative to the median merit value, which was 0.020. Like in Run I, preferential sampling of points at

low temperatures and reagent mixtures with high molar excesses of C_{60} is seen. However, in Run IVb the cluster of sub-median merit values is closer to the 'spine' of the bounding cage, representing longer residence times arising from low reagent flow rates. The best point found during Run IVb had a merit value of -0.166, and a composition of 0.101 (X_0), 0.784 (X_1), 0.110 (X_2), and 0.004 (X_3).

Already it can be seen that we have successfully encoded our product specification using the soft constrained optimisation. The composition with the lowest merit value found during the normal SNOBfit optimisation, Run IVa, showed that the amount of X_3 can be minimised easily by moving to conditions with short residence time, low temperature, and more C_{60} in the reagent mixture. However, these conditions, exemplified by the best point, will have only a very small amount of X_2 and an abundance of unconverted X_0 . By using a soft constraint on the overall conversion to X_1 and X_2 , Run IVb was able to find a best point much closer to our product specification, almost exactly matching it.

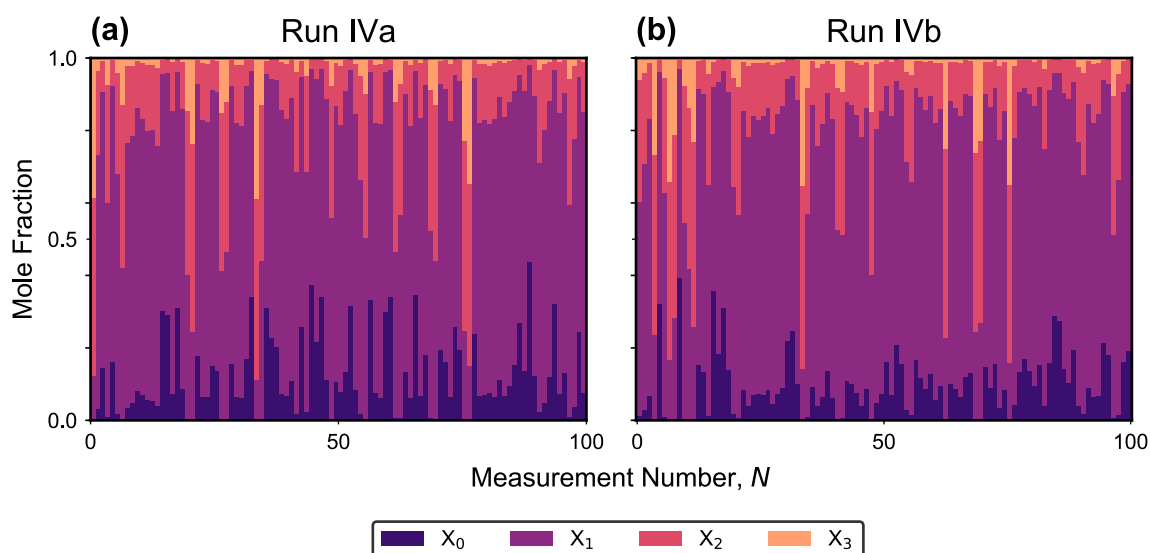


Figure 4.13: The composition of the product at every measurement during the unconstrained and constrained SNOBfit optimisations in Run IVa (a) and Run IVb (b).

The mole fractions of each of the four derivatives are plotted against measurement number, N , in Fig. 4.13a for Run IVa. Once again, the substantial fluctuations can be seen in the mole fractions between consecutive measurements due to the shift from local searching at the beginning of each batch of recommended measurements, to global searching at the end of each batch. A similar picture is seen for the constrained optimisation, Run IVb, in Fig. 4.13b, albeit with the area representing $[X_0]$ appearing to be lower on average than in Run I. From this it can be seen that both optimisations preferentially sampled reaction conditions that produced almost no X_3 - indicated by its near absence from both plots - moderate amounts of X_2 , and large amounts of X_1 , while Run IVb also preferentially selected points with low X_0 as well.

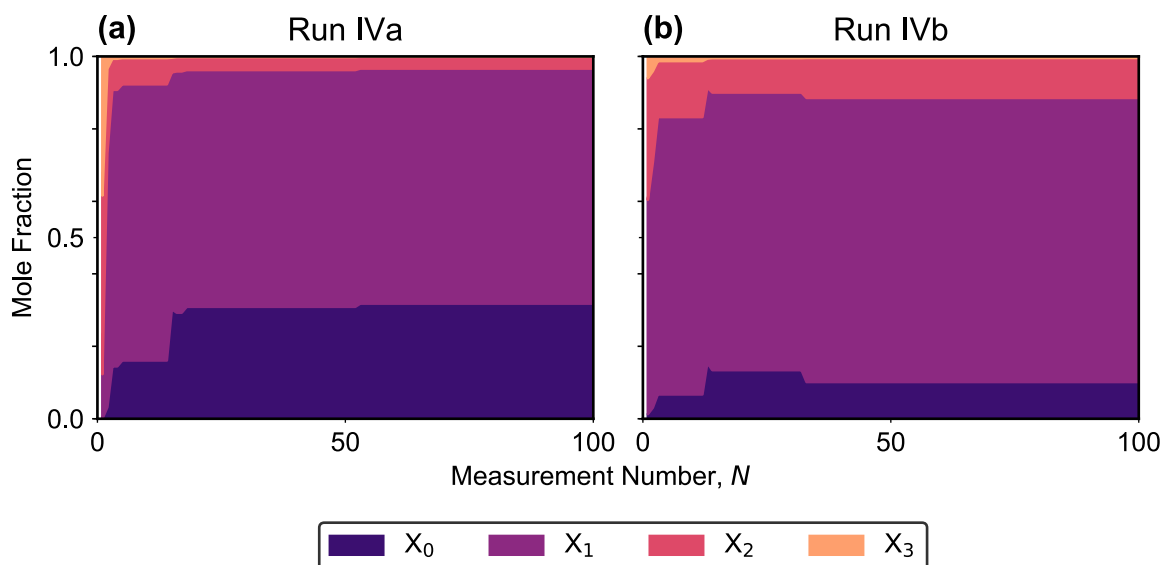


Figure 4.14: The composition of the product at the best result as it is updated during the SNOBFit and soft constrained SNOBFit optimisations of Run IVa (a) and Run IVb (b).

Another view of this is provided by Fig. 4.14a and 4.14b, where the mole fractions of the four derivatives at the best result to date are shown against N , for Run IVa and Run IVb respectively. For Run IVa, improvements, i.e. reductions, in the amount of X_3 in the product occurred at $N = 5, 15, 18$, and 53 , with the mole fraction of X_3 falling from an initial value of 0.384 at the first measurement to the lowest overall value of 0.001 by measurement 53 . Conversely, the mole fraction of unreacted X_0 increased from 0.000 at measurement 1 to 0.318 at measurement 53 . From this it is clear that the reduction in X_3 seen during Run I was achieved at the cost of decreasing the overall conversion from C_{60} . Looking at the plot for Run IVb, changes in the best result occurred at $N = 3, 13, 14$, and 33 , with $[X_3]$ taking values of $0.013, 0.006, 0.005$, and 0.004 , and $[X_0]$ taking values of $0.067, 0.155, 0.134$, and 0.101 . This once again shows that using a soft constrained optimisation has successfully achieved the product specification: an initial drop in $[X_3]$ was accompanied by a corresponding increase in $[X_0]$, like in Run I, but the further reductions in $[X_3]$ also gave reductions in $[X_0]$. As such, Run IVb was able to reduce $[X_3]$ without also reducing the conversion from C_{60} .

That the soft constrained optimisation was able to do this by preferentially exploring regions of the parameter space with low $[X_0]$ is further demonstrated in Fig. 4.15a and 4.15b. In both plots, the mole fraction composition of each product sampled has been ranked by the amount of X_3 in it. Both show what would be expected; as the amount of the third-order derivative in the product decreased, the amount of the second-order derivative increased slightly, before decreasing as the amount of first-order, then zeroth-order derivatives increased. The unconstrained SNOBFit run minimising $[X_3]$ (Fig. 4.15a) and the constrained SNOBFit run (Fig. 4.15b), appear broadly similar. However, careful

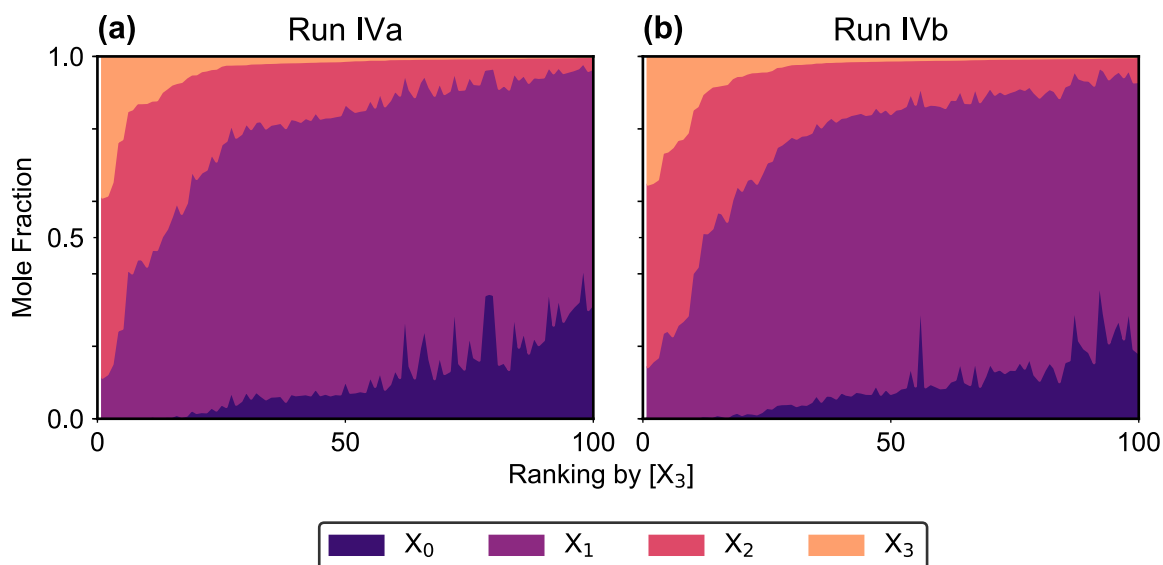


Figure 4.15: Comparison of the composition of the product formed at each measurement in the SNOBfit and soft constrained SNOBfit optimisations of Run IVa (a) and Run IVb (b), ranked by the mole fraction of X_3 in them, with the lowest at 0 and the highest at 100.

inspection reveals that the area representing the mole fraction of X_3 is slightly larger in the plot for Run IVb, and the area for $[X_0]$ is smaller than in Run IVa. Overall this demonstrates that Run IVb preferentially probed the parameter space in areas that led to high conversion from C_{60} .

More insight into how the optimisations ran, and the difference between just trying to minimise the amount of a detrimental product and attempting to encode a product specification into the optimisation, can be seen in Fig. 4.16. For Run IVa, Fig. 4.16a shows the successive merit values throughout the optimisation, which (as well as once again showing the optimisation alternating between low merit value local searches and high merit value global searches) demonstrates the algorithm preferentially searching areas of the parameter space that produce low merit values. This is further reinforced by Fig. 4.16b, which is a histogram of the merit values sampled in Run IVa showing a large peak in the distribution at the lowest function values, and a tail that drops off quickly with some small amount of points sampled with merit values up to just under 0.4. In fact, in Run IVa more than 80% of the sampled data points had a merit value of 0.05 or below, compared to the maximum measured merit value of 0.389.

Similar plots are shown for Run IVb in Fig. 4.17, but this time as well as showing the successive values for the merit function, $f^{\text{soft}}(\mathbf{x})$, the values for the scaled objective function, $f^*(\mathbf{x})$, and the penalty function, $p(\mathbf{x})$, are shown in Figs. 4.17a(i), 4.17b(i), and 4.17c(i). The corresponding histograms are displayed in Figs. 4.17a(ii), 4.17b(ii), and 4.17c(ii). As would be expected, like the unconstrained SNOBfit optimisation, the soft constrained optimisation also preferentially sampled areas of the parameter space that resulted in low merit values. In this case, these low merit values were achieved through

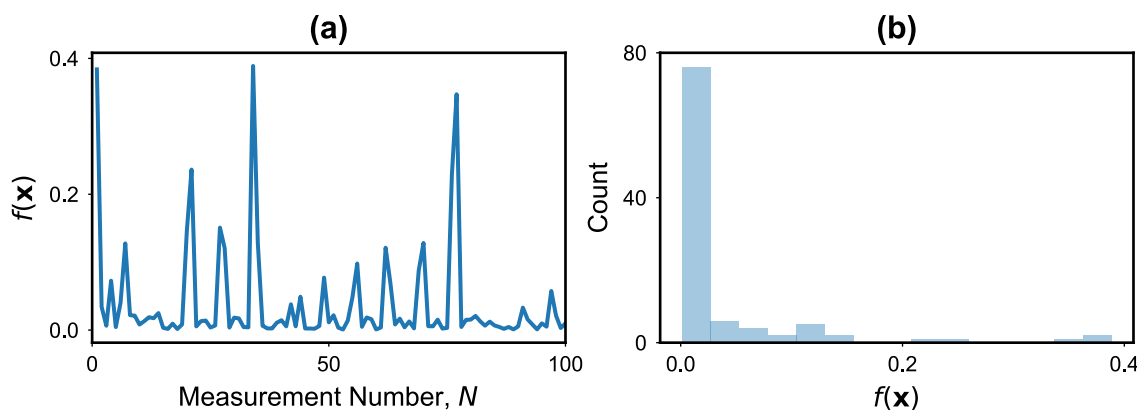


Figure 4.16: The merit value, $f(\mathbf{x})$ (a), and the distribution of values sampled for $f(\mathbf{x})$ (b) during the course of Run IVa

a combination of low $f^*(\mathbf{x})$ and low $p(\mathbf{x})$ values. While the $f^*(\mathbf{x})$ values had a range from -0.227 to 0.887, they were strongly biased towards negative values, meaning that most points identified by the algorithm had a low objective function value, corresponding to a smaller amount of X_3 , than the first identified feasible point $\mathbf{x}_0$ (found at the first measurement). The histogram for $p(\mathbf{x})$ is heavily skewed towards 0, showing that the algorithm had a strong bias towards points that either fully or nearly satisfied the constraint: out of 100 measurements, 46 had a penalty of 0, so were fully feasible. As such, this demonstrates that by having an algorithm that is preferentially samples the parameter space for points that minimise both the objective function and a penalty function that encodes constraints on the product, the whole product specification can be optimised optimised, rather than one part of it over another.

In Figs. 4.18a(i) and 4.18a(ii), chromatograms of the best products found during Run IVa and Run IVb are displayed. Comparing these two chromatograms, it is evident that $[X_{12}] = [X_1] + [X_2]$ was much higher in Run IVb, at almost 0.9, than in Run IVa, which was 0.68 at its best point. Also shown in Fig. 4.18a(ii) is the chromatogram of the best feasible point found during the Run IVb. The fully feasible and semi-feasible best point chromatograms are very similar, almost lying exactly on top of each other. This implies that they have very similar product compositions. Indeed, the best feasible point had a third-order derivative mole fraction of 0.007 and a combined mole fraction $[X_{12}]$ of 0.905, compared to 0.004 and 0.894 for the best semi-feasible point. From this comparison it is clear that the reduction in $f^{\text{soft}}(\mathbf{x})$ was achieved by a decrease in the objective function target, $[X_3]$, at the cost of a small violation of the constraint on $[X_{12}]$.

This is in contrast to the previous examples of soft constrained optimisations, where there was a large difference in composition between the best feasible points and the points with the overall lowest $f^{\text{soft}}(\mathbf{x})$. In those optimisations, the reduction in $f^{\text{soft}}(\mathbf{x})$ between the best feasible and the best semi-feasible points was accompanied by a large improvement in the objective function, and a significant violation of the constraint function. For

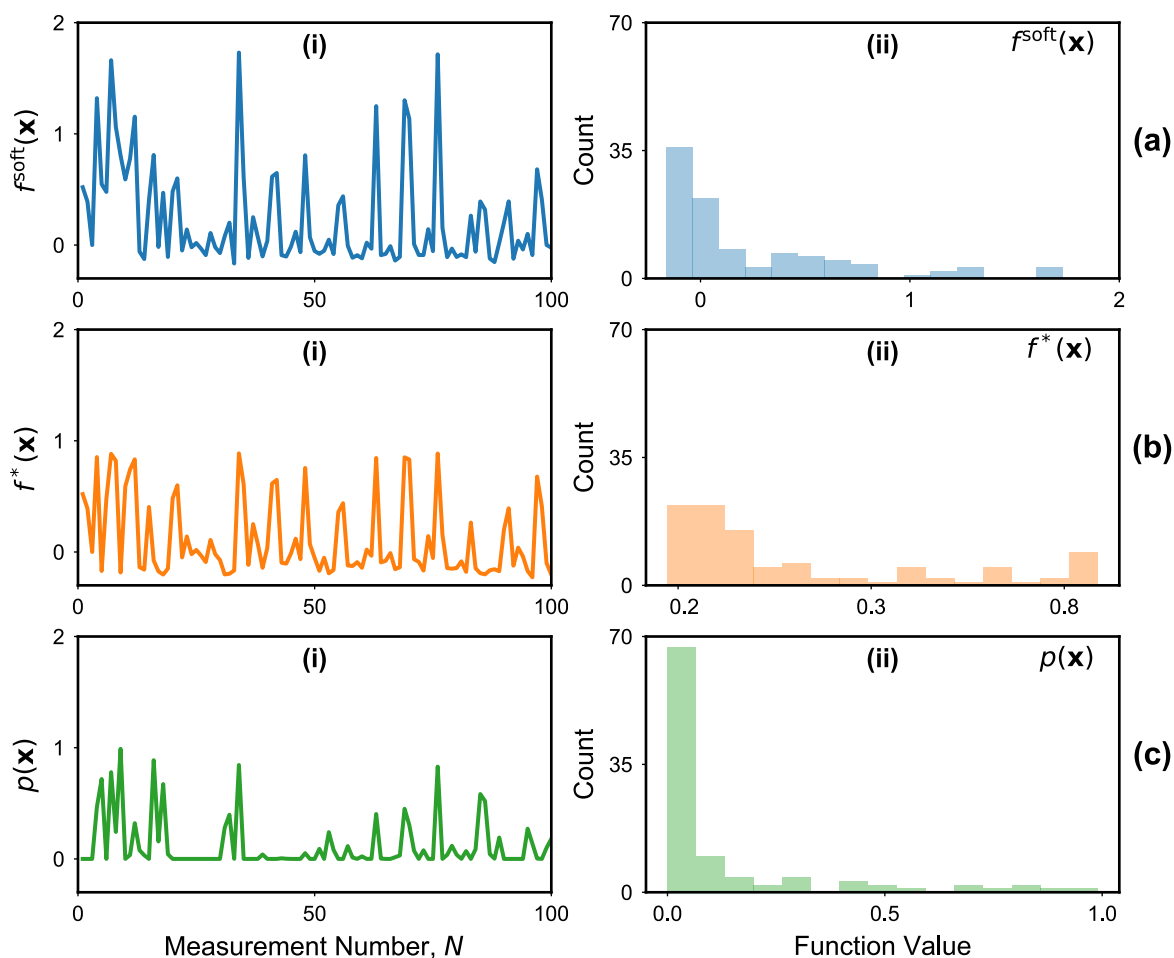


Figure 4.17: The value of $f(\mathbf{x})$ (a(i)), $f^*(\mathbf{x})$ (b(i)), and $p(\mathbf{x})$ (c(i)) measured during Run IVb, and the corresponding distribution of values.

Run IVb, only a small violation of the constraint function was needed to achieve a slightly better merit value, as the merit value of the best fully feasible point already had virtually no X_3 . Any larger increase in the objective function value for Run IVb would have been accompanied by a much larger violation of the constraints. This suggests that unlike the examples of soft optimisations run when investigating the effect of the σ parameter, the optimisation in Run IVb was well formulated.

Run IVb shows how a product specification can easily be encoded into a soft constrained optimisation using the soft merit function described here. However, it used a fairly simple specification, placing only a constraint on the combined mole fraction of $[X_{12}]$. A more interesting product specification is one where constraints are placed on the ratios of the different fullerene derivatives. For instance, some application may require just the singly-substituted derivative. As seen from the previous results, high amounts of X_1 can also result in significant amounts of X_2 , and so the most cost effective solution in this case might be one that has a ratio of X_1 to X_2 above a particular value.

A third optimisation, Run IVc, was carried out for this new product specification

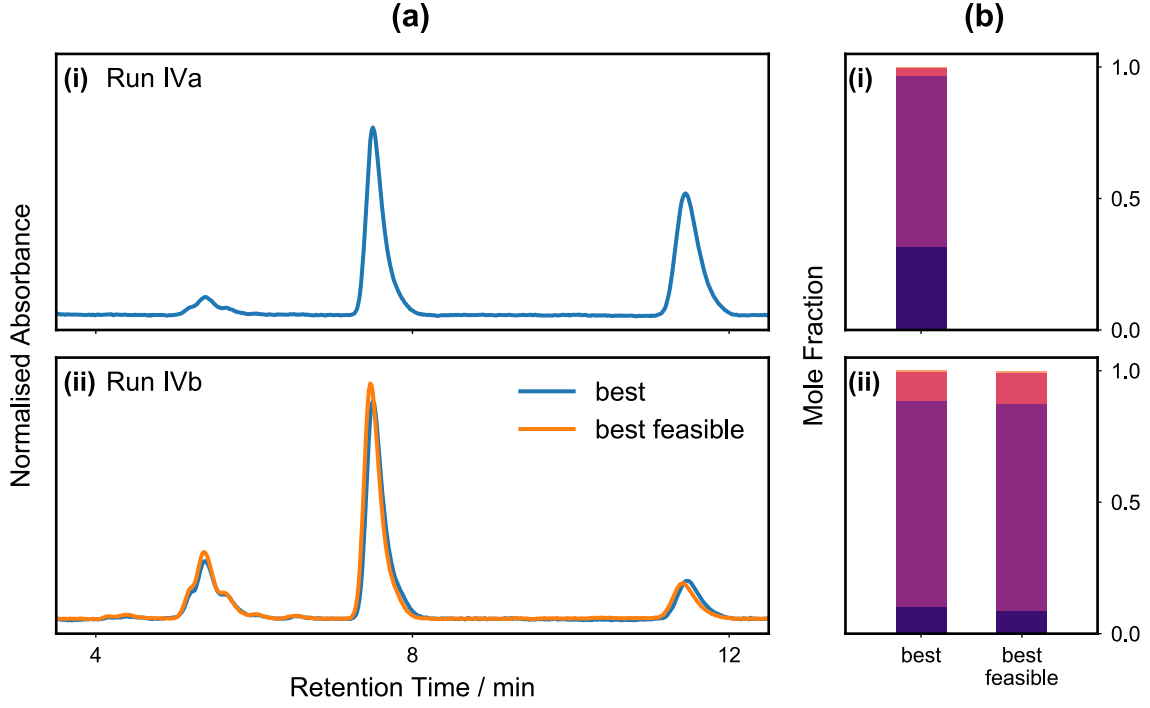


Figure 4.18: Chromatograms of the best and best feasible points identified during the SNOBFit and soft constrained SNOBFit optimisations in Run IVa (a(i)) and Run IVb (a(ii)), alongside corresponding bar charts (b) showing the composition at these points.

using the same objective and constraint functions as in Run IVb, but applying an additional constraint on the ratio of X_1 to X_2 , $\mathcal{R}_{12} = [X_1]/[X_2]$. This was encoded into the optimisation as in Eqns. 4.9.

$$f(\mathbf{x}) = [X_3] \quad (4.9a)$$

$$F_1(\mathbf{x}) = 0.9 < [X_2] + [X_1] \quad (4.9b)$$

$$F_2(\mathbf{x}) = 4 < \mathcal{R}_{12} \quad (4.9c)$$

By constraining $\mathcal{R}_{12}$ to be above 4, it was hoped that a product rich in X_1 , but with high conversion from C_{60} and low in X_3 could be achieved. Once again, the optimisation was run for 100 measurements, and the conservative value of 0.3 was used for σ .

During Run IVc both a best feasible and an overall best point were identified. The best point was identified at measurement 26, with mole fractions of 0.129, 0.776, 0.091 and 0.004. These imply an $\mathcal{R}_{12}$ value of 8.5:1 and $[X_{12}]$ of 0.867. The best feasible point, on the other hand, was identified at measurement 99, and had mole fractions of 0.085, 0.778, 0.128, and 0.009, with implies an $\mathcal{R}_{12}$ value of 6.1:1 and $[X_{12}]$ of 0.906. The corresponding

chromatograms can be seen in Fig. 4.19a(i). From these, it is clear that a small violation in the constraint on $[X_{12}]$ allowed a small reduction in $[X_3]$ from 0.009 to 0.004, while the constraint on $\mathcal{R}_{12}$ remained satisfied at both points.

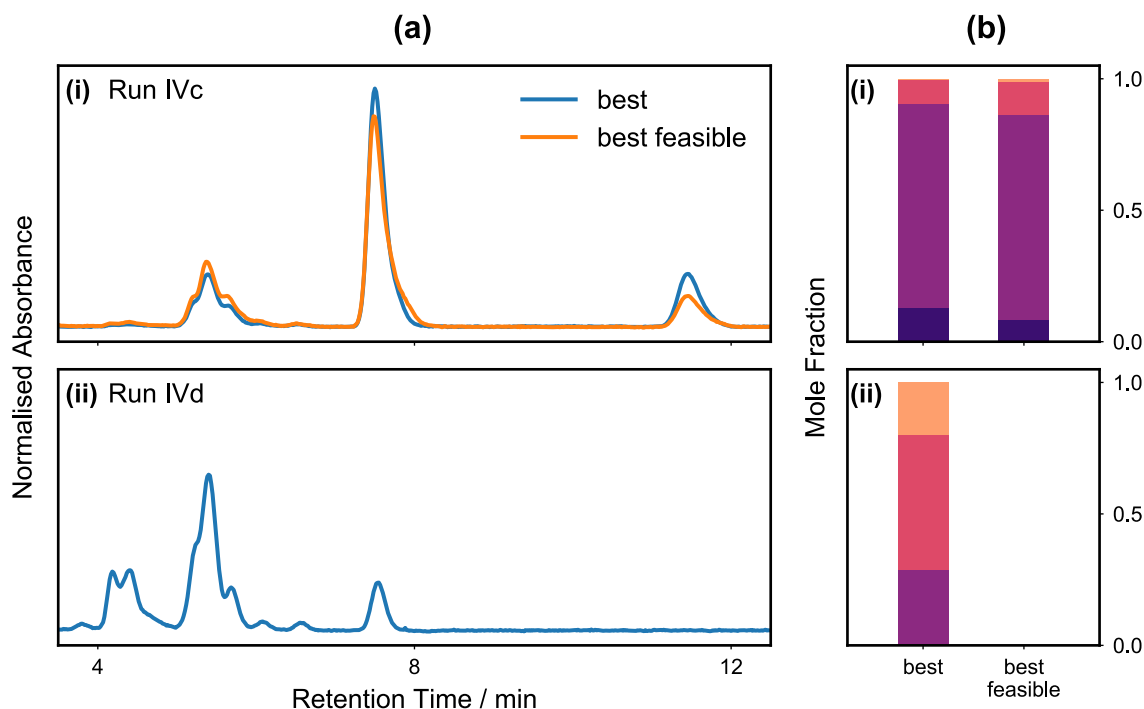


Figure 4.19: Chromatograms of the best and best feasible points identified during the soft constrained SNOBfit optimisations in Run IVc (a(i)) and Run IVd (a(ii)), alongside corresponding bar charts (b) showing the composition at these points.

Despite the successful encoding of the product specification rich in X_1 in the optimisation in Run IVc, getting a product with lots of X_1 is not an especially difficult task. In fact the previous optimisation, Run IVb, without the constraint on $\mathcal{R}_{12}$ managed to get a best point with $\mathcal{R}_{12}$ of 7.4, and a best feasible point of 6.7. As such, comparable results were gained without encoding the full specification in the optimisation. A much harder problem, and one more relevant to the application of *o*-xylenyl derivatised fullerenes, is to create a product with high conversion from C_{60} , low concentration of X_3 , and is rich in the most useful derivative X_2 .

The product specification can be encoded using the same objective and constraint functions as in Run IVc, but with an upper limit of 0.5 on $\mathcal{R}_{12}$ (shown in Eqns. 4.10) in order to get a product that is has at least twice as much X_2 as X_1 .

$$f(\mathbf{x}) = [X_3] \quad (4.10a)$$

$$F_1(\mathbf{x}) = 0.9 < [X_2] + [X_1] \quad (4.10b)$$

$$F_2(\mathbf{x}) = \mathcal{R}_{1,2} < 0.5 \quad (4.10c)$$

Another optimisation was carried out, Run IVd, using this formulation of the problem, 0.3 as σ , and the same boundaries as before. Demonstrating that this indeed was a more difficult problem, no feasible point was found during Run IVd. As no feasible points were found, the penalty function $p(\mathbf{x})$ did not reach zero, instead having a range of values from 0.280 to 1.999. Of the 100 measurements that the optimisation was run for, 50 were partial violations, with 36 partially violating just the constraint on $[X_{12}]$, and the other 14 partially violating both constraints. Of the 50 full violations, 2 violated only the constraint on $[X_{12}]$, 26 violated just the constraint on $\mathcal{R}_{12}$, and the remaining 22 violated both. These violations highlight the difficulty of getting a product with a high ratio of $[X_2]$ to $[X_1]$ while also suppressing $[X_3]$. That over half of the infeasible points were full violations of the constraint on this ratio, but none of the semi-feasible points were partial violations of just this constraint alone, suggests that the greatest difficulty was in satisfying the constraint on $\mathcal{R}_{12}$.

The chromatogram of the best point from Run IVd, found at measurement 15 with a merit value -0.212, is shown in Fig. 4.19a(ii). From the sizes of the peaks, it looks like the goal of getting an excess of X_2 over X_1 has been achieved, albeit with a significant amount of X_3 . The mole fractions of the derivatives at this best point were 0.001, 0.288, 0.510, and 0.201 respectively. This gave the ratio $\mathcal{R}_{12}$ of 0.564, partially violating the constraint by 0.064, and the combined mole fraction $[X_{12}]$ as 0.798, partially violating its constraint by 0.102. Given how close the the ratio constraint was to its bound, it seems like the algorithm attempted to satisfy this constraint by violating the constraint on the combined mole fraction, and allowing a high fraction of the third-order derivative in the product. Like the optimisation run earlier with a very low value for σ , in Run IVd the penalty function appears to have a greater weight in the soft merit function than the scaled objective function. This means that the algorithm could more easily reduce the overall merit value by reducing the violation of the constraints than by reducing the objective function, thus leading to a poorly minimised objective function with a large amount of X_3 in the product.

In order to visualise the difficulty of simultaneously achieving a high ratio of second to first order derivatives while also suppressing formation of the third-order derivative, illustrative chromatograms acquired during Run IVd with $[X_3]$ decreasing from 0.400 to 0.100 are shown in Fig. 4.20a. From these chromatograms it is clear that as $[X_3]$ is reduced, $[X_2]$ decreases at a slow rate, while $[X_1]$ increases dramatically. At no point do the chromatograms show a point where both $[X_3]$ and $[X_1]$ are low, while $[X_2]$ is high.

This trade-off between the ratio of second to first order derivatives and the amount

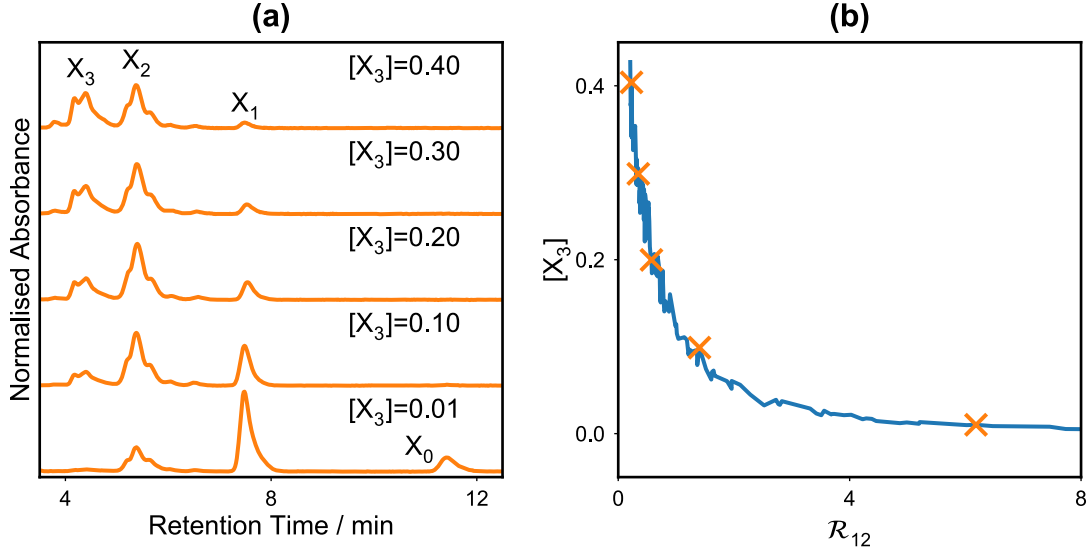


Figure 4.20: The trade-off between the ratio of $[X_1]:[X_2]$, $\mathcal{R}_{12}$, and $[X_3]$ (a) from the results of Run IVd, and chromatograms chosen with 0.01, 0.10, 0.20, 0.30, and 0.40 $[X_3]$ (b), marked on the trade-off curve.

of third order derivative is further illustrated by the trade-off curve of $[X_3]$ and $\mathcal{R}_{12}$ in Fig. 4.20b, plotted from the results of Run IVd. The points used to plot the chromatograms have been marked on the curve, which shows $[X_3]$ decreasing as $\mathcal{R}_{12}$ increases. Taking the previously applied limit of X_3 composing 10% of the product as an acceptable upper limit, the trade-off curve shows that $\mathcal{R}_{12}$ can not be below 1.5.

Another optimisation was carried out, Run IVe, armed with this new information. The same σ value of 0.3 and the same bounds as before were applied, but this time the optimisation was formulated with the objective and constraint functions in Eqns. 4.11.

$$f(\mathbf{x}) = [X_3] \quad (4.11a)$$

$$F_1(\mathbf{x}) = 0.9 < [X_2] + [X_1] \quad (4.11b)$$

$$F_2(\mathbf{x}) = \mathcal{R}_{1,2} < 1.5 \quad (4.11c)$$

This time an initial feasible point, $\mathbf{x}_0$, was found an measurement 21, which turned out to be both the point with the lowest merit value and the feasible point with the lowest merit value. The chromatogram for this point is shown in Fig. 4.21a(i). The mole fractions were 0.007, 0.535, 0.373, and 0.085, giving values of 0.909 and 0.143 for $[X_{12}]$ and $\mathcal{R}_{12}$. As this feasible point is in close agreement with the trade-off curve, it is clear that the optimisation in Run IVd was carried out with unsatisfiable constraints.

However, the data acquired during that run was still useful in that it was used to identify a more reasonable limit for the ratio constraint, that would allow the amount of X_3 to be minimised as well.

Although this formulation of the product specification eventually lead to an acceptable result, with a product relatively rich in X_2 , that had little to no X_3 and X_1 , it required two optimisations to get there. In a real world situation this might not be a satisfactory solution, given the time and material costs associated with running each optimisation. Thinking a bit more about the product specification - that we want to maximise the amount of X_2 while minimising the amount of X_3 and maximising conversion from X_1 - can lead to an alternative way to encode the product specification into a soft constrained optimisation. Given the cascadic nature of the reaction, ensuring that $[X_{12}]$ is above 0.9 should result in both a high conversion from C_{60} and a mole fraction of X_3 below 0.1, as if this constraint is satisfied only 10 % of the product can be either X_0 or X_1 . Furthermore, in Run IVd it was unclear what an achievable upper limit on $\mathcal{R}_{12}$ was, but for a product rich in X_2 it should be as low as possible. As such, an alternative formulation of the product specification is to minimise $\mathcal{R}_{12}$, while placing a constraint on the combined mole fraction of $[X_{12}]$, as in Eqns. 4.12.

$$f(\mathbf{x}) = \mathcal{R}_{1,2} \quad (4.12a)$$

$$F(\mathbf{x}) = 0.9 < [X_{1,2}] \quad (4.12b)$$

In effect, this would be finding the optimum point along the trade-off curve for a given amount of X_3 , used as the constraint upper limit.

A final optimisation was carried out using this objective and constraint function, Run IVf, and the same σ and bounds as before, for 100 measurements. This successfully found both a best point and a best feasible point, the chromatograms for which are shown in Fig. 4.21a(ii). The composition of the best feasible point was 0.007, 0.527, 0.378 and 0.087 for $[X_0]$, $[X_1]$, $[X_2]$, and $[X_3]$ respectively. This gave a value of 1.393 for $\mathcal{R}_{12}$, and 0.906 for $[X_{12}]$. This is in reasonable agreement, if not slightly better, than the best feasible point for Run IVe, which had an alternative formulation of the problem. The mole fractions of the fullerene derivatives at the best point were 0.004, 0.382, 0.458, and 0.157. Clearly the result with the lowest merit value has an unsatisfactory amount of X_3 , with the combined mole fraction $[X_{12}]$ dropping to 0.840, but accompanied by a favourable reduction of $\mathcal{R}_{12}$ to 0.833.

In these six optimisations, the advantage of using a soft constrained optimisation to encode the specification of a desired product have been demonstrated. A number of different product compositions were achieved by selecting an appropriate metric to minimise,

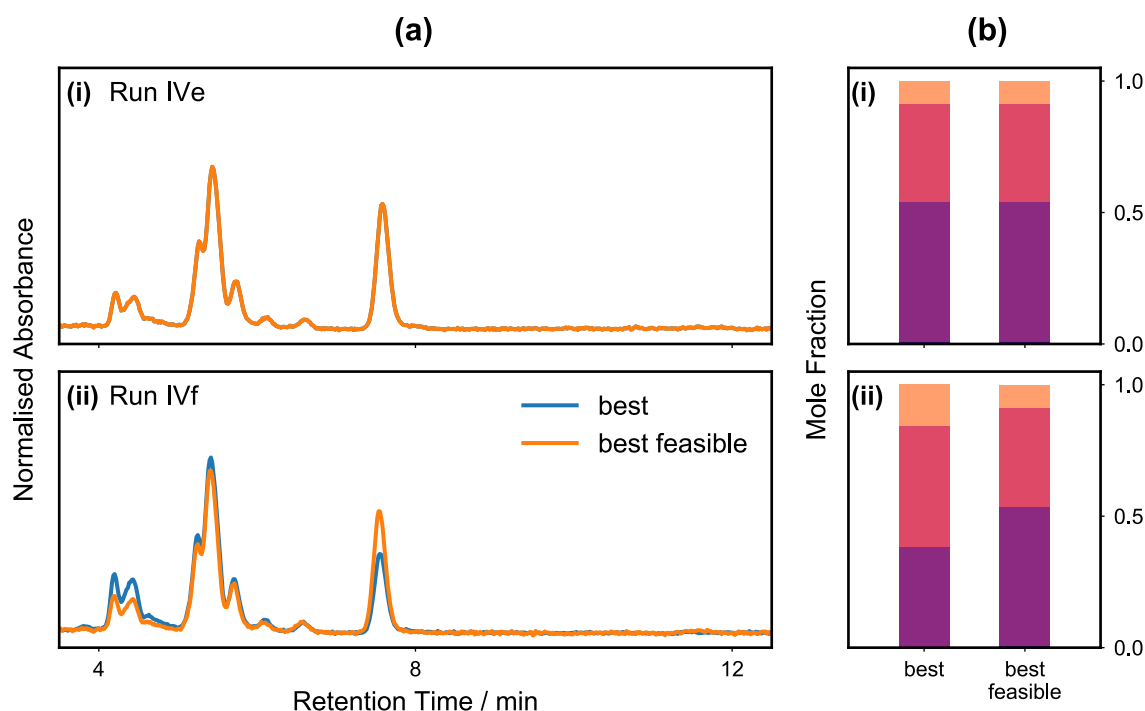


Figure 4.21: Chromatograms of the best and best feasible points identified during the soft constrained SNOBFit optimisations in Run IVe (a(i)) and Run IVf (a(ii)), alongside corresponding bar charts (b) showing the composition at these points.

e.g. the mole fraction of the third-order derivative, and applying sensible constraints to other metrics, e.g. the ratio of the first-order derivative to the second-order derivative. It was seen that although the selection of these metrics is fairly natural, without needing to choose any fudge factors, some thought was required to choose a sensible set of objective and constraint functions. With careful consideration of the product specification, a product relatively rich in X_2 , with high conversion from C_{60} , and low in X_3 was achieved.

4.3 Conclusions

This chapter focused on the development of an intuitive method for encoding a product specification into a self-optimising reactor. Building on previous work using an automated flow reactor for self-optimisation of the synthesis of *o*-xylenyl derivatised fullerene adducts, an alternative method for carrying out multi-objective chemical optimisations was developed as a result of the difficulties experienced when using a weighted-sum approach. In this approach, the problem was formulated as a soft constrained optimisation, and the SNOBFit algorithm was used to optimise a merit function that was a combination of a scaled objective function and a penalty function describing the violation of constraint functions. The feasibility of using this approach was investigated, and a number of optimisations were run using it and changing the softness of the constraints.

Armed with more knowledge about using constrained optimisations to optimise the trade-off between competing requirements of a chemical synthesis, this approach was then used to try and gain products that matched particular specifications. By carefully considering what quantity should be minimised, and which should be constrained, a number of different product specifications were produced. First, a specification requiring as little X_3 as possible, while ensuring over 90 % of the product was converted from C_{60} , was successfully attempted. After trialling a few more specifications, an optimisation that maximised the amount of X_2 relative to X_1 , while ensuring no more than 10 % of the product was X_3 was successfully run. In doing so, it was shown that a product specification could be encoded into a soft constrained optimisation in a couple of different ways, and get similar results. Furthermore, it was found that the most effective way to optimise a product specification with no prior experiments is to use the objective function to optimise the quantity that has no acceptable value, either because it needs to be as high or low as possible or the acceptable value is difficult to define. As many constraint functions as needed can then be used to specify the rest of the product. Used in this way, the method described here for running soft constrained optimisations becomes an intuitive and powerful tool for optimising chemical syntheses with a number of products.

CHAPTER 5

**Synthesis of Iron Oxide
Nanoparticles in a Three-Phase
Droplet Reactor**

5.1 Introduction

The following chapter departs from the focus on automation and optimisation to describe preliminary work on the flow synthesis of iron oxide nanoparticles. The aim of this was to take advantage of the increased control and more uniform reaction conditions provided by the large surface area reaction channels⁵, to develop a tunable synthesis of magnetic iron oxide nanoparticles.

5.1.1 Magnetic Properties of Iron Oxide Nanoparticles

Iron oxide nanoparticles exhibit magnetic properties distinct from bulk iron oxide that make them useful for a range of different applications. In bulk iron oxide there are areas of magnetisation that are separated from each other by energetic barriers, or walls, called domains. The magnetic moments of these domains are typically disordered, so bulk iron oxide has no net magnetisation and therefore does not exhibit any magnetic properties. However, as the size of a piece of iron oxide is reduced, the energy needed to create walls between domains becomes greater than the energy reduction from creating a new domain. Below a particular size (128 nm for iron oxide¹⁰⁹), the magnetisation of the material will be in a single domain. As such, iron oxide nanoparticles smaller than 128 nm are single domain magnets.

The magnetisation of a single domain magnet is aligned along a single direction, to minimise energy, called the easy axis. The magnetisation can flip between different easy axes if an energy barrier, called the anisotropy energy, is overcome. This barrier depends on the volume of the nanoparticle, as well as its shape and material. For spherical nanoparticles of a particular material, the anisotropy energy (and therefore the barrier to flipping the easy axis) increases linearly with volume.

Applying an external magnetic field can align the magnetic moments in a sample of nanoparticles in a single direction. If the anisotropy energy of the nanoparticles is high enough, their magnetic moments will remain aligned even when the external magnetic field is removed. In this case, the nanoparticles are said to be ferromagnetic, and the magnetisation of the sample shows hysteresis as the strength and direction of an external magnetic field is changed (see Fig. 5.1a). However, if the anisotropy energy of the nanoparticles is less than the ambient thermal energy ($k_B T$), the magnetic moments of the nanoparticles will become unaligned quickly as they flip between different easy axes. In this case, the nanoparticles are said to be superparamagnetic, and there is no hysteresis in how their magnetisation changes with an applied magnetic field (see Fig. 5.1b).

Iron oxide nanoparticles are typically superparamagnetic, because although they are single domain magnets as a result of their relatively large single domain limit (128 nm compared to 15 nm for iron nanoparticles¹⁰⁹), they have a small anisotropy constant (-

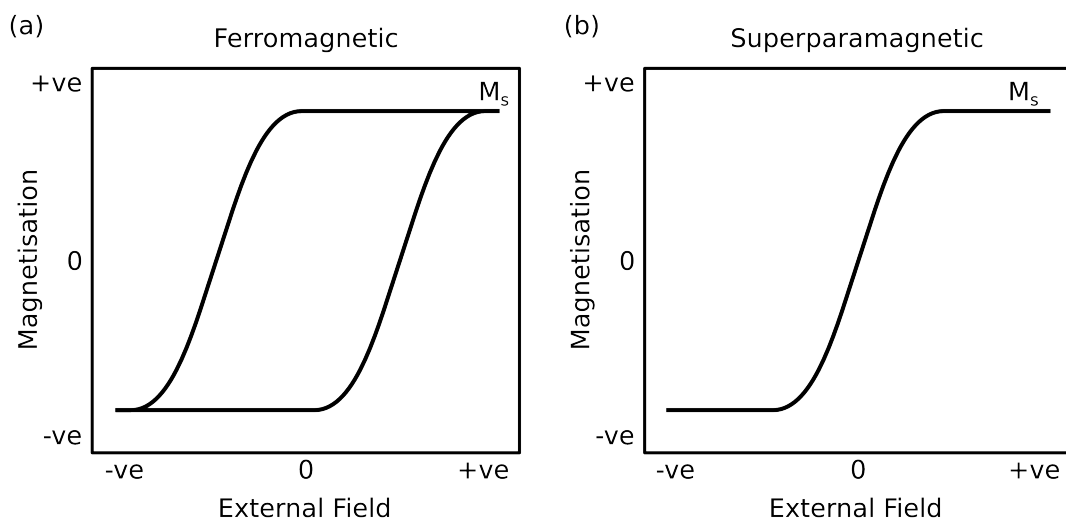


Figure 5.1: Diagrams of the change in magnetisation of magnetic nanoparticles in an external field as the field strength changes for (a) ferromagnetic nanoparticles, showing hysteresis, and (b) superparamagnetic nanoparticles, showing no hysteresis. The saturation magnetisation has been marked on both as M_s

$1.1 \times 10^4 \text{ Jm}^{-3}$ compared to $4.8 \times 10^4 \text{ Jm}^{-3}$ ¹⁰⁹), which makes the barrier to flipping the direction of magnetisation low.

As well as determining their magnetic behaviour, the size of magnetic nanoparticles can also affect the magnitude of their magnetisation. The saturation magnetisation is the maximum magnetisation of a material, determined by the total number of magnetic moments in the material. Magnetic moments in the surface of magnetic nanoparticles, especially oxides, can be aligned off-axis, or canted¹¹⁰. This spin canting leads to a reduction in the saturation magnetisation of the nanoparticles. The surface area to volume ratio of a nanoparticle increases with decreasing diameter, which causes an increase in the spin canting effect. Therefore, smaller nanoparticles generally have a lower saturation magnetisation.

The superparamagnetic properties of iron oxide nanoparticles, as well as their high saturation magnetisation, means that they can respond quickly to external magnetic fields. This is useful for a number of biomedical applications^{111–123}. For example, applying an alternating magnetic field can cause the direction of magnetisation in a sample of iron oxide nanoparticles to rapidly switch, converting the magnetic energy into thermal energy. This allows localised heating, which can be used in magnetic fluid hyperthermia therapy to destroy or inhibit the growth of cancer cells^{116,120–122}. Similarly, their response to an external magnetic field means iron oxide nanoparticles can be used as contrast agents for magnetic resonance imaging (MRI), as their induced magnetic fields can enhance spin relaxation of protons^{111,124}. Their high magnetisation also makes them useful in catalysis, where they have been used as supports for catalytic materials to allow them to be magnetically recovered^{125–127}.

The size and size distribution of iron oxide nanoparticles in a sample is important for these applications. To take advantage of the high saturation magnetisations, the nanoparticles need to be smaller than the single domain limit. Similarly, the nanoparticles need to be below the superparamagnetic limit for applications that require superparamagnetism. However, if the nanoparticles are too small, surface effects such as spin canting can reduce the saturation magnetisation. Furthermore, a large distribution of nanoparticle sizes can also reduce the overall saturation magnetisation, as some nanoparticles may be larger than the single domain limit, or may be much smaller than the rest of the sample.

Different sizes may also be required for different applications. For example, in MRI there are two proton spin relaxation processes that can be examined: transverse relaxation (T_2), due to spin-spin dephasing, and longitudinal relaxation (T_1), due to realignment of the proton spins with the external magnetic field. The high magnetisation of superparamagnetic iron oxide nanoparticles means that they are very good T_2 MRI contrast agents^{109,111}. However, very small iron oxide nanoparticles, with small magnetisation due to strong spin canting effects, have been shown to be very efficient T_1 MRI contrast agents^{109,111}. Different nanoparticles sizes are also needed for a range of biomedical applications when targeting different tissues, as well as in catalysis to support different catalytic materials.

From this it is clear that a high degree of control is required over the synthesis of iron oxide nanoparticles in order to produce high quality samples with a uniform size distribution. Furthermore, a size-tunable synthesis is needed to make nanoparticles that are tailored for a particular application.

5.1.2 Iron Oxide Nanoparticle Synthesis

A number of synthetic routes have been reported for iron oxide nanoparticles including coprecipitation^{128–131}, high temperature decomposition (thermolysis) of metal-organic precursors^{102,103,132–134}, and polyol syntheses^{135,136}. Of these, coprecipitation and thermolysis are the most amenable to a flow reactor, as usually all reactants are completely dissolved before reacting, and both methods have been used previously for the flow synthesis of other types of nanoparticles^{22,28,29,33,49,77}. Despite requiring high temperatures, thermolytic syntheses may be preferable because of the superior control they provide over nanoparticle size and size distribution, and phase purity¹³⁷.

Flow Synthesis of Iron Oxide Nanoparticles

There have been a few reports to date of magnetic iron oxide nanoparticles being synthesised in flow reactors. Most of these use an aqueous coprecipitation synthesis^{41,138–140}. For example, Frenz *et al* used a droplet-flow microreactor to synthesise iron oxide nanoparticles by the coprecipitation of Fe^{II} and Fe^{III} salts under basic conditions¹³⁸. Of particular

note is that they introduced their two reagent mixtures, aqueous solutions of iron salts and ammonium hydroxide, as two droplet streams. This produced a flow of pairs of reagent droplets that they then fused together by applying an electric field across a section of the reaction channel. In doing so they were able to rapidly form iron oxide nanoparticles, while avoiding reactor fouling caused by uncontrolled reagent mixing. In a more recent example, Kumar *et al*⁴¹ used a more conventional droplet flow reactor, where a single droplet flow was generated. Like Frenz *et al* they used the coprecipitation route, and were able to synthesise superparamagnetic iron oxide nanoparticles with a saturation magnetisation of 58 emu/g - higher than the typical values of 30-50 emu/g reported for the coprecipitation route elsewhere, but not as high as the values of 65 to 82 emu/g obtained by syntheses using the thermal decomposition route^{102,103}.

A flow synthesis of iron oxide nanoparticles using the thermal decomposition of iron oleate has been reported by Glasgow *et al*¹⁴¹. Unlike the other flow syntheses discussed, they were able to tune the size of nanoparticles produced. By varying both the temperature and the mass ratio of iron salt to capping ligand in the precursor solution, they made samples of iron oxide nanoparticles with average diameters from 7 to 16 nm. However, the authors note that there was no conclusive trend in particle size when varying the ratio of iron salt to capping ligand, and so they demonstrated only rudimentary control over particle size. Furthermore, their nanoparticles had very low saturation magnetisations (around 0.6 emu/g).

This last example was carried out in a continuous flow reactor. Reactor fouling, by deposition of nanoparticles on the reaction channel walls, is a major concern for nanoparticle synthesis in flow^{22,33}. Although Glasgow *et al* do not mention any reactor fouling, it would likely have been a problem if they had operated their reactor for longer. The flow reactors reported by Frenz *et al*¹³⁸ and Kumar *et al*⁴¹ both used a droplet flow regime to prevent nanoparticle deposition and the subsequent reactor fouling, as has been done for the flow synthesis of a range of other nanomaterials^{6,19,22,24,28,32,35,38,43,46,57,77,142}.

Of the reported flow syntheses of iron oxide nanoparticles discussed above, only the report by Glasgow *et al* demonstrated any control over the size of the nanoparticles. However, the nanoparticles that they produced had much lower saturation magnetisations than other syntheses, for example the droplet flow synthesis reported by Kumar *et al*⁴¹. Furthermore, the thermal decomposition of iron oleate that they used resulted in iron oxide nanoparticles capped with oleic acid. This meant that they were not water-soluble, unlike the nanoparticles produced in the flow syntheses using the coprecipitation route. This is beneficial for many medical and bioimaging applications, where solubility in (aqueous) biological media is a pre-requisite. Although Dr. Kritika Kumar did go on to demonstrate control over the diameter of water-soluble iron oxide nanoparticles in their PhD thesis¹⁴³, this was only over the range of 2.5 to 7 nm. Larger diameter iron oxide nanoparticles are needed to get the highest saturation magnetisations.

In this chapter I describe the adaptation of a droplet-based microreactor reported by Nightingale *et al*⁵⁷ to the synthesis of water soluble iron oxide nanoparticles using a synthetic route reported by Zhen Li *et al*¹⁰². This reaction was thought to be a good fit for a flow synthesis because it uses a relatively small number of reagents - $\text{Fe}(\text{acac})_3$ is used as both the source of iron and the reducing agent, as the acetyl acetonate decomposes, and 2-pyrrolidone is used as both the solvent and the nanoparticle capping ligand (see Fig. 5.2). As 2-pyrrolidone is the capping agent, the synthesis gives water-soluble nanoparticles with minimal work up and no need for any ligand exchange steps.

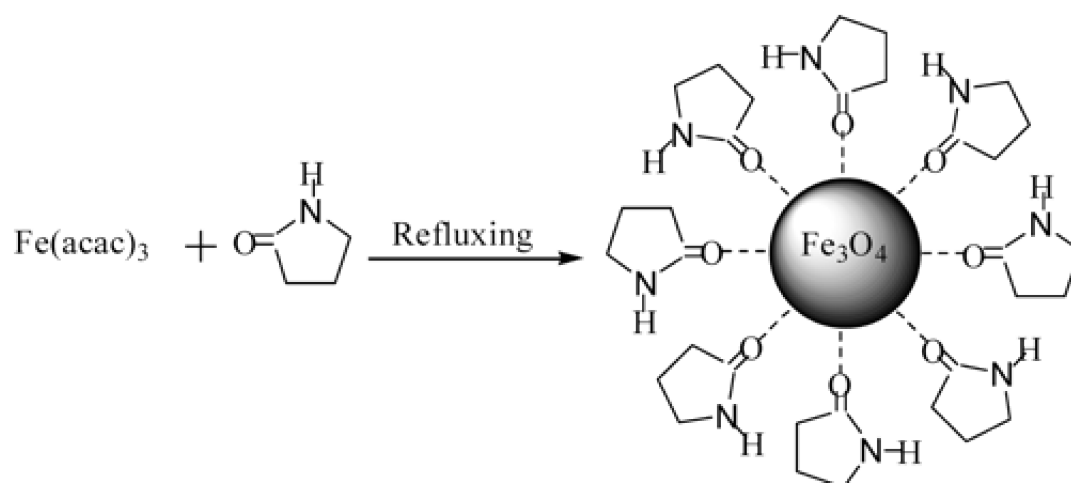


Figure 5.2: Synthesis of water-soluble iron oxide nanoparticles by the thermal decomposition of $\text{Fe}(\text{acac})_3$ in refluxing 2-pyrrolidone, where the 2-pyrrolidone acts both as reaction solvent and nanoparticle capping ligand. Reprinted with permission from Zhen Li *et al*¹⁰². Copyright 2002 American Chemical Society.

Zhen Li *et al*¹⁰² had shown this one-pot synthesis to give magnetite crystals with an average diameter of 5 nm when 2.0 mmol $\text{Fe}(\text{acac})_3$ was refluxed in 2-pyrrolidone for 10 min. Using a seed-mediated method and the same reagents, they were also able to produce larger crystals with a diameter of 11 nm, suggesting that the average nanoparticle diameter could be tuned using the number of seed particles and reaction time. It was therefore hoped that converting this synthesis to a flow system would give more control over conditions for greater tunability, as well as higher throughput than traditional synthesis.

Bulk synthesis trials carried out by Dr. Thomas Phillips had given a good yield of magnetic iron oxide nanoparticles. However, when the synthesis was adapted to a droplet flow system fouling became a large issue. This chapter describes attempts to develop a more stable, fouling-free synthesis of iron oxide nanoparticles with control over temperature and reaction time, through control of droplet residence time.

5.1.3 Three-Phase Flow Synthesis of Iron Oxide Nanoparticles

A three-phase flow regime has previously been used to prevent fouling caused by coalescence of reagent droplets in a segmented flow reactor, and so was seen as a possible solution to the previous problems encountered in adapting the synthesis to flow. In a three-phase system, as well as using an immiscible carrier fluid, an inert gas is co-injected along with the reaction mixture (Fig. 5.3, top). The carrier fluid works as in a two-phase segmented flow regime, causing droplets of reagent mixture to form, and creating a barrier between these droplets and the reaction channel wall. The addition of a gas creates a barrier between the droplets, as they are now separated by gas slugs, and thereby prevents them from coalescing.

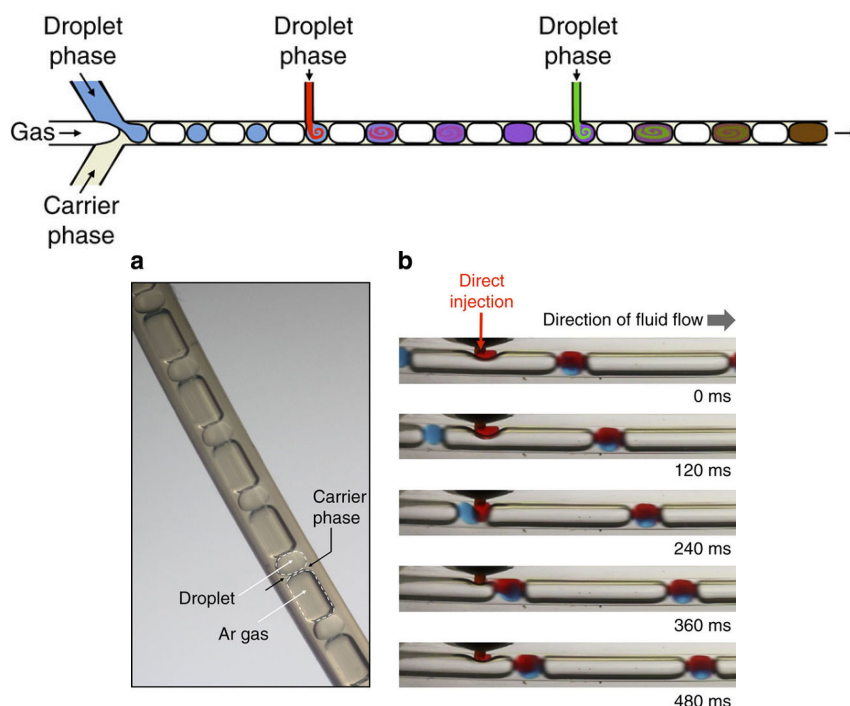


Figure 5.3: A schematic of three-phase flow formation and downstream addition of reagents (top) with photographs of the three-phase droplet flow showing the separation of phases (bottom, a) and the addition of fresh reagent mixture (bottom, b). Reproduced from Nightingale *et al*⁵⁷

A small number flow reactors have been reported that use a three-phase flow regime to prevent droplet coalescence, and therefore reactor fouling. For example, Duraiswamy and Khan used a three-phase flow regime to synthesise thin gold shells on silica nanoparticles for plasmonic applications⁵⁵. More recently, a flow reactor for the large scale synthesis of palladium nanoparticles has been developed by Wong *et al*⁴⁹, using a three-phase flow regime to be able to operate for long periods of time without any reactor fouling. A multi-channel three-phase flow reactor has also been reported from the same group, by Yap *et al*¹⁴⁴, that took advantage of the gas-phase to carry out a hydrogenation reaction.

A three-phase flow reactor has also been used by Nightingale *et al* for a multi-step

nanoparticle synthesis⁵⁷. As well as separating the droplets of reaction mixture, the slugs of gas had a high surface tension that prevented injection of fresh reagent mixture into the space between droplets (Fig. 5.3, bottom). While other methods of introducing new reagents into droplets have been reported, such as droplet merging¹⁴⁵ or direct injection into a two-phase flow^{146,147}, the use of a three-phase flow regime allowed more consistent and reliable dosing and prevented any partial mixing or new droplet formation.

Nightingale *et al* used their three-phase flow reactor to synthesise large volume (~ 43 nm³) CdSe quantum dots by a seed-mediated route. A wide range of nanoparticles have been reported using the seed-mediated route^{148–151}, where a batch of small nanoparticles are first synthesised and then further grown in a second stage of the synthesis. This method has been demonstrated to offer exceptional control over the size and shape of nanoparticles, by using two separate reactions for their nucleation and growth. By forming nanoparticle seeds as the first stage in a multi-step flow synthesis, Nightingale *et al* were able to exhibit excellent control over the formation of CdSe nanocrystals.

In their synthesis of iron oxide nanoparticles, Zhen Li *et al* were able to produce larger (11 nm) nanoparticles by using a seed-mediated synthesis. Therefore, a three-phase flow reactor was developed for the synthesis of iron oxide nanoparticles to prevent the fouling experienced previously when using a two-phase droplet flow synthesis, as well as creating the possibility of using a seed-mediated synthesis by multi-step addition of the reagent mixture.

5.1.4 Development of a Three-Phase Flow Synthesis of Iron Oxide Nanoparticles

Previous attempts at developing a three-phase synthesis of iron oxide nanoparticles using the 2-pyrrolidone route had been made by Dr. Thomas Phillips. These used a flow reactor design based on the three-phase flow reactor reported by Nightingale *et al*. However, the reactor suffered from an unstable gas and liquid flow. At periodic intervals, the velocity of the gas would surge. It was thought that increasing the back-pressure in the system would provide resistance against these surges, and thereby stabilise the system. To this end, I developed a three-phase flow reactor based on this work, but incorporating a back-pressure regulator into the system at the outflow of the reaction channel.

Three-phase systems represent a specific problem for back-pressure regulators. Most commercially available back-pressure regulators for flow reactors use a membrane or ball-and-spring to apply pressure. As such, they are only suitable for liquids. Other back-pressure regulators are available for gases, but I am not aware of any low-cost solution for mixtures of gas and liquid. For this reason, a bespoke back-pressure regulator, in the form of a pressurised vessel that the reaction channel pumps into, was designed and built by Dr. James Bannock. A similar back-pressure regulator has been reported by Deadman

*et al*¹⁵² that used a pressurised collection vessel to apply a high back-pressure to a slurry-forming reaction, thereby avoiding the blockages that would occur with a conventional back-pressure regulator. Their back-pressure regulator was designed to accommodate superheated conditions, and so used materials that could withstand high temperatures and very high pressures. Therefore, the back-pressure regulator designed by Dr. James Bannock differs from theirs in its application to a three-phase gas-liquid flow and in the simplified design used due to the lower pressures required by the synthesis of iron oxide nanoparticles.

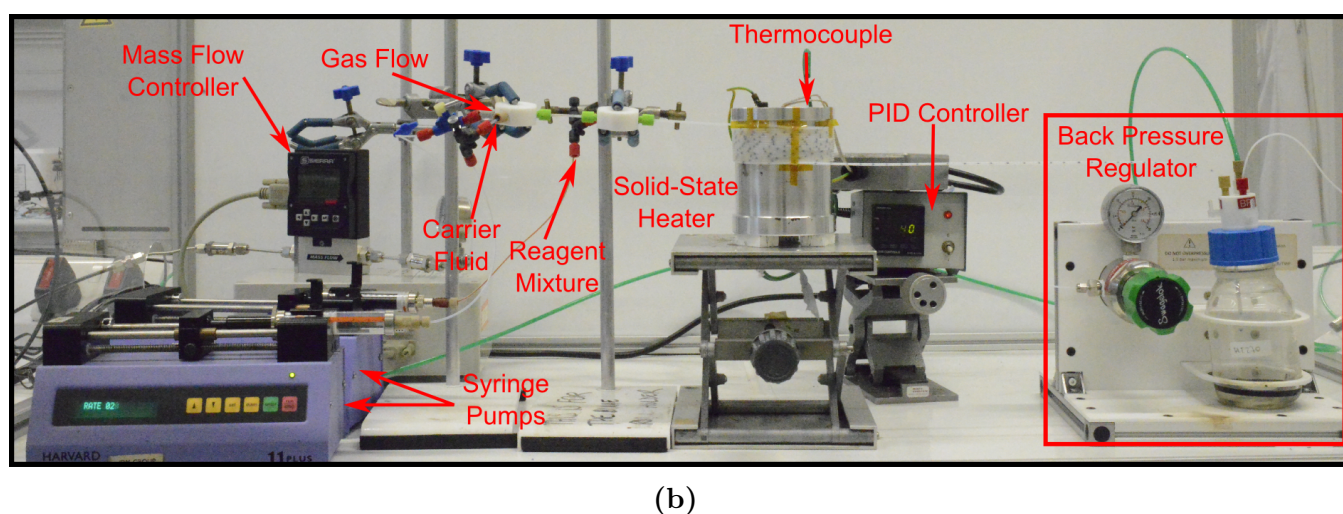
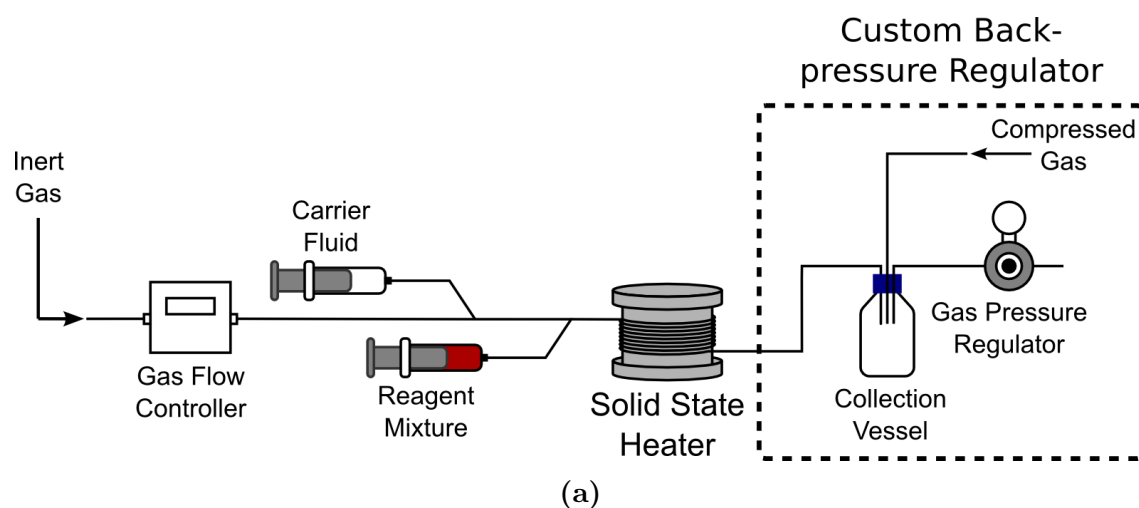


Figure 5.4: (a) Schematic of the three-phase flow reactor for synthesising magnetic iron oxide nanoparticles and (b) a photograph of the system with each of the components labelled.

The design of the flow reactor, shown as a schematic in Fig. 5.4a and as a photograph in Fig. 5.4b, was based on the three-phase flow reactor reported by Nightingale *et al*. The flow rate of gas into the reactor was controlled by a mass flow controller (MicroTrak 101, Sierra Instruments), while the flow rates of the inert carrier fluid and reagent mixture were controlled using syringe pumps (Pump 11+, Harvard Apparatus). The gas and carrier fluid were introduced through a two-inlet PTFE droplet generator, used in previous work

reported by Bannock *et al*^{7,19,42,43} and Nightingale *et al*²⁴, a cross-section of which is displayed in Fig. 5.5a. The three-phase flow was generated across two stages, using two separate droplet generators, as can be seen in Fig. 5.5b. This gave more control over the flow than generating it over one step by decoupling reagent droplet formation and gas/carrier-fluid slug formation. In the first stage, the carrier fluid was introduced to a gas flow to produce a gas-liquid slug flow. The gas flow was connected to this first droplet generator using a small-diameter glass capillary that protruded into the tubing connected to the outlet of the droplet generator. By doing so, it was hoped that the added back-pressure of the glass capillary would produce a more stable gas flow. This also allowed the formation of the segmented flow of gas and carrier fluid to be directly inspected through the transparent tubing. The second stage was the addition of the reagent mixture to the gas-liquid slug flow through a second droplet generator, forming the three-phase droplet flow.

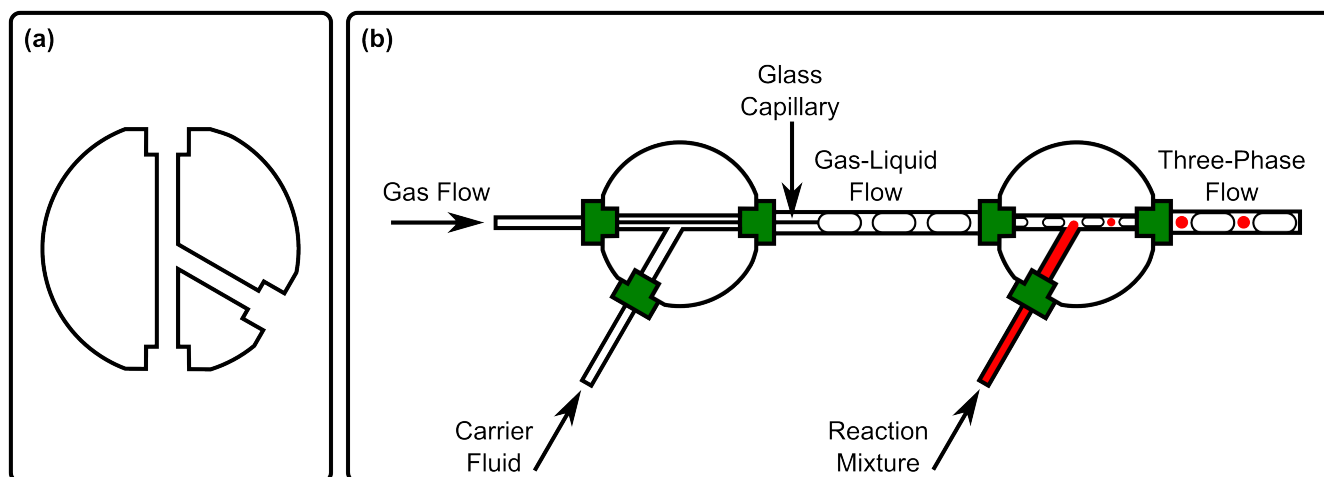


Figure 5.5: Schematic diagrams of (a) a cross-section of a two-inlet PTFE droplet generator showing the spaces for connecting tubing with microfluidic nuts, and (b) the construction of a three-phase flow using sequential droplet generators, with the carrier fluid being introduced to a gas flow first to set up a gas-liquid flow, and then the reagent mixture being introduced.

The outlet of this second droplet generator was connected to the reaction channel, which was helically wrapped around a solid-state heater and fastened using high-temperature adhesive tape. The solid-state heater was made from an aluminium block with a lip machined into it at either end similar to the heater used in the automated flow reactor developed for the synthesis of *o*-xylenyl fullerene derivatives, described in Chapter 3. A single cartridge heater was placed in a hole drilled into the centre of it, and a K-type thermocouple was put in a hole drilled near the surface of the block. Ceramic feet were attached to the bottom. The temperature was controlled using a PID controller (Series 32A, Love Controls), which controlled the current supplied to the cartridge heater and measured the temperature using the thermocouple.

The outlet of the reaction channel was attached to a collection bottle with an integrated

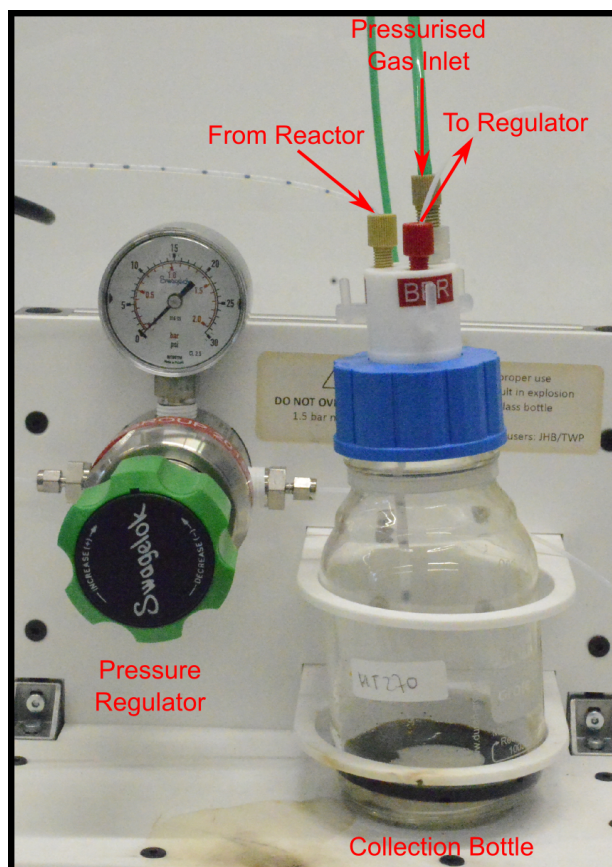


Figure 5.6: Photograph of the custom-designed back pressure regulator, consisting of a pressurised collection bottle with two inlets, one for the reactor outlet and one to allow the bottle to be pressurised with compressed air. The outlet of the bottle was connected to a gas pressure regulator, which could be used to control the pressure in the collection bottle. This back pressure regulator was designed and built by Dr. James Bannock.

back-pressure regulator, custom designed and built by Dr. James Bannock. This was made from a collection bottle with a lid that had three tap-controlled inlets. The outlet of the reactor was attached to one of these. Another was connected to a supply of compressed air, while the third was connected to the open air through a gas pressure regulator. The back-pressure regulator was operated by keeping the flow reactor and gas regulator taps open, and opening the tap connected to the compressed air to pressurise the collection bottle. Once a particular pressure had been achieved, the compressed air inlet tap was closed. The pressure inside the bottle could then be released, or increased to a certain extent, by using the gas regulator.

Through systematic experimentation, a successful way of controlling the system was developed. The reactor was operated by first heating it to the desired temperature. To ensure a stable gas-liquid flow, the gas and carrier fluid syringe pumps were then started at the desired flow rates. The back-pressure was slowly increased by gradually charging the pressurised collection vessel, waiting to observe the change in gas-liquid flow, and charging the vessel with more gas if necessary. Just as the pressure began to get high enough to

stabilise the gas-liquid flow, the reagent syringe pump was started. More back-pressure was then carefully applied if necessary.

It was important to prevent any reagent mixture from touching the reaction channel walls while the reactor was hot, to avoid fouling. This was one reason for obtaining a stable gas-carrier fluid flow before introducing the reagent mixture. Similarly, after using the reactor, the reaction channel was cleared of reagent mixture by flowing carrier fluid through it.

As the pressure was increased using the back-pressure regulator, there was a danger of the carrier fluid, due to its high viscosity, flowing back into the tubes connecting the gas and reagent mixture to the droplet generators. This would lead to contamination of these tubes that was hard to remove. For this reason, taps were fitted to the gas and reagent lines to close them off if necessary, and the back-pressure was increased carefully as described above.

By starting the reactor with no reagent mixture flowing, fouling of the reactor from the inevitable unstable flow at the beginning of an experiment was avoided. Careful increases in the back-pressure of the system were necessary as it was much harder to control decreases in back-pressure than increases, so over-pressuring the collection vessel was undesirable. Finally, the reagent flow was introduced just as or before the gas-liquid flow was becoming stable, as it increased the total back-pressure in the system, sometimes by an undesirable amount.

As the liquid flow rates were controlled using syringe pumps, these could be specified in conventional volumetric flow rate units of microlitres per minute (μLmin^{-1}). The gas flow rate, controlled by a mass flow controller, was specified in units of standard cubic centimetres per minute (scc/min). A standard cubic centimetre is the volume (in cm^3) that an amount of gas would occupy under standard conditions. As the flow reactor was not operating under standard conditions, the volumetric flow rate of gas into the reaction channel was not directly determined by the value set on the mass flow controller, but was also dependent on the temperature and pressure of the reaction channel. Therefore, the average residence time of the reaction mixture droplets could not be calculated from the liquid and gas flow rates set on the syringe pumps and mass flow controller, and instead had to be directly measured. A method for measuring the droplet residence time had not been developed when the synthesis was being adapted to the flow reactor, and so the residence time was unknown for the first experiments described in this chapter.

5.2 Synthesis of Iron Oxide Nanoparticles in 2-Pyrrolidone

The synthesis of iron oxide nanoparticles by the 2-pyrrolidone route, reported by Zhen Li *et al*¹⁰², was adapted to the three-phase flow reactor described above. A stable flow with well-spaced droplets was attained using a gas flow rate (Q_{gas}) of 0.4 scc/min, a carrier fluid flow rate (Q_{carrier}) of 100 μLmin^{-1} , and a reagent flow rate (Q_{reagent}) of 25 μLmin^{-1} . It was thought the risk of reactor fouling would be lower if the droplets were well separated from each other, and small enough that there was a substantial amount of carrier fluid between them and the reaction channel walls.

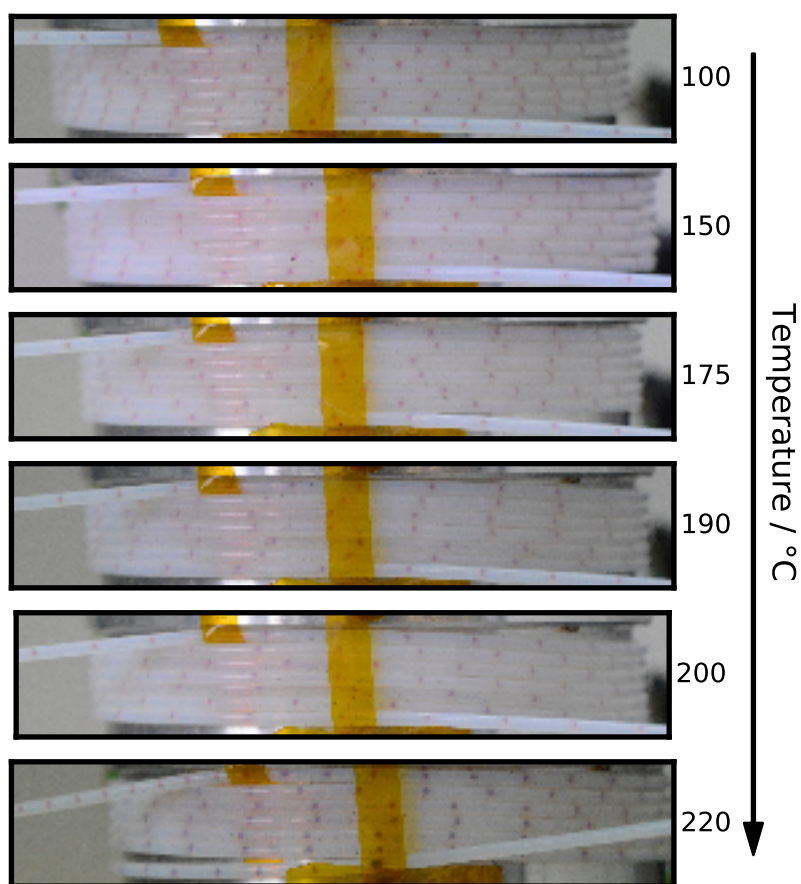


Figure 5.7: Photographs of reaction mixture droplets flowing through the heated reaction channel at temperatures increasing from 100 °C to 220 °C, showing no appreciable change in the colour of the droplets as they pass through the reactor, indicating no nanoparticle formation.

Flow rates of $Q_{\text{gas}} = 0.4$ scc/min, $Q_{\text{carrier}} = 100$ μLmin^{-1} , $Q_{\text{reagent}} = 25$ μLmin^{-1} , and back-pressure of 3 psi were used for all experiments.

The bulk synthesis¹⁰² was carried out at reflux, at a temperature of 245 °C for 2-pyrrolidone. Although the degradation temperature of PTFE is 350 °C, it begins to deteriorate (soften) at 260 °C, causing the reaction channel tubing to deform. For this

reason I placed an upper limit of 240 °C on the temperature of the three-phase reactor. Given the improved heat transfer in flow compared to batch syntheses³³, due to the high surface area to volume ratio of the reaction channel, it was hoped that the synthesis could be carried out at a lower temperature than the bulk synthesis.

The three-phase flow synthesis was carried out using the previously identified stable flow rates, at increasing temperatures of 100 °C, 150 °C, 175 °C, 190 °C, 200 °C, and 220 °C. The back-pressure was adjusted so that it was constant at all temperatures, at 3 psi. A stable three-phase droplet flow was first achieved at 100 °C using the procedure described above, and the temperature was increased subsequently in ascending order. Once the droplet flow had stabilised at each temperature, the reactor was left to run for at least 15 min before assessing whether nanoparticles had formed. Although it would have been preferable to randomise the order that the temperatures were tested in, the main purpose of these preliminary tests was to find the lowest possible temperature that the synthesis could be run at. It was not intended to be a rigorous study of the effect of temperature on the synthesis of iron oxide nanoparticles.

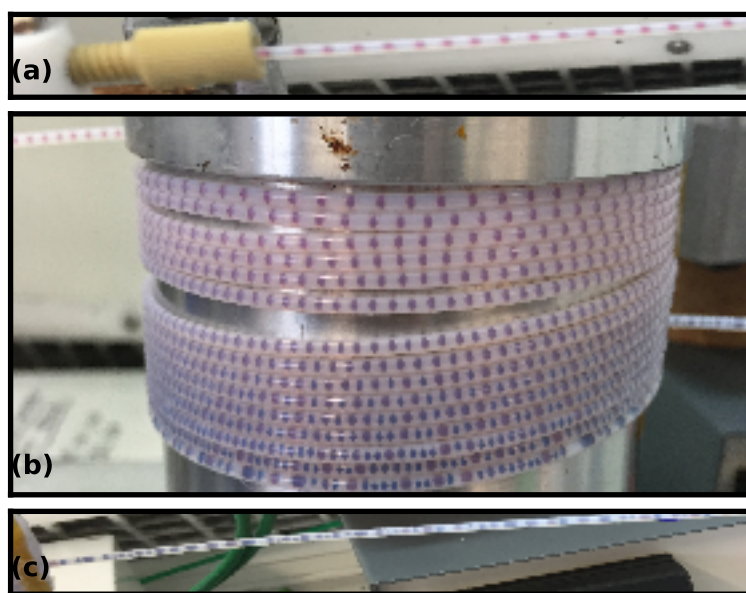


Figure 5.8: Photographs of (a) the inlet, (b) the heated reaction channel, and (c) the outlet of the reactor for the conditions $Q_{\text{gas}} = 0.4$ scc/min, $Q_{\text{carrier}} = 100 \mu\text{Lmin}^{-1}$, $Q_{\text{reagent}} = 25 \mu\text{Lmin}^{-1}$, and back-pressure of 4 psi. A change of colour from bright red to dark purple/black can be seen in the droplets moving from the inlet to the outlet. Regular droplet can be seen at the inlet, in contrast to the irregular droplets caused by coalescence during heating.

The photographs in Fig. 5.7 suggest that particles did not form at any of the temperatures tested. The droplets were the same red colour at the inlet of the heated reaction channel as at the outlet, with no observable darkening in colour as would be expected from the formation of iron oxide nanoparticles, suggesting the reactor was not hot enough, or the droplets were not heated for long enough. To ensure that a reaction occurred, it

was decided that both the temperature and droplet residence time should be increased.

The three-phase synthesis was tried again at the maximum reactor temperature of 240 °C. As the aim at this stage of experimentation was simply to achieve nanoparticle formation in the flow reactor without any fouling, the length of the reaction channel was increased from 1.51 m to 3.77 m. This allowed for longer residence times without having to reduce the spacing between the reagent droplets or slowing the gas flow too much, which could have given more opportunity for fouling.

The conditions used for the synthesis were $Q_{\text{gas}} = 0.4 \text{ scc/min}$, $Q_{\text{carrier}} = 100 \mu\text{Lmin}^{-1}$, and $Q_{\text{reagent}} = 25 \mu\text{Lmin}^{-1}$, with a back-pressure of 4 psi. Photographs of the reagent droplets at the reactor inlet, heated reaction channel, and reactor outlet taken during this experiment are shown in Fig. 5.8(a-c) respectively. The droplets at the reactor inlet were evenly spaced and regularly sized, indicating a stable flow had been attained. Although the droplet residence time was not measured for this experiment, it was qualitatively observed to increase as a result of the increase in back-pressure from 3 to 4 psi. In the reaction channel, the droplets changed from the bright red of the reagent mixture to a much darker colour at the bottom of the reactor. This suggested that iron oxide nanoparticles were successfully formed in the flow reactor at the higher temperature of 240 °C.

However, it can be seen that as the droplets progressed through the heated section of the reaction channel, the spacing between them became smaller. Towards the end of the heated section, the size of the droplets became irregular, with smaller, darker droplets interspersed with larger seemingly unreacted ones. At the reactor outlet, the size and spacing of the droplets was very irregular. After running the reactor at these conditions for 40 min there were signs of fouling. Letting the reactor cool and emptying the reaction channel confirmed that it had badly fouled, with large deposits of iron oxide on the walls of the reaction channel, shown in Fig. 5.9.

The irregular droplet size and spacing, and subsequent fouling of the reactor channel, suggested that the synthesis of iron oxide nanoparticles by thermal decomposition of $\text{Fe}(\text{acac})_3$ would be difficult to implement with our three-phase flow system. Despite using a specially designed back-pressure regulator to gain a stable gas-liquid flow velocity, and good regular droplet formation at the reactor inlet, the reactor still fouled. It was observed that at a particular point in the reaction channel, shortly after entering the heated section when the droplets would have reached the reaction temperature, the droplets started to move to one side, bringing them into contact with the channel wall. This may have been a consequence of the surface energy of the reaction solvent, 2-pyrrolidone, changing as it neared its boiling point, causing it to have a higher affinity to the PTFE tubing than the PFPE carrier fluid. Another possible explanation is that despite the increased back-pressure, the solvent began to boil and the resultant out-gassing pushed the droplets to the side of the reaction channel. As well as causing deposition of iron oxide nanoparticles,

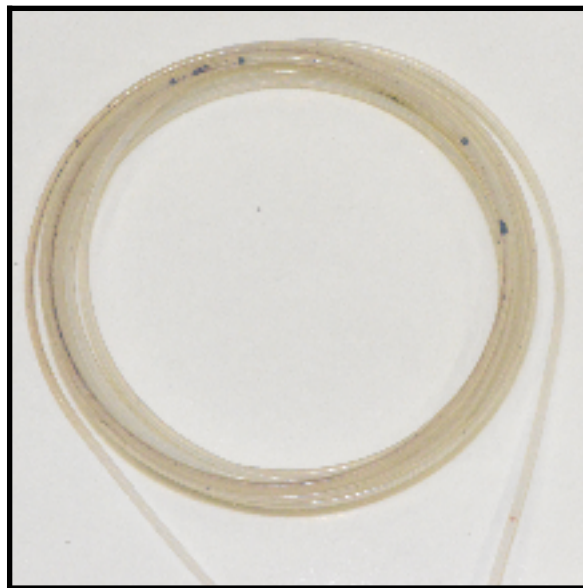


Figure 5.9: A photograph of the PTFE tubing used as the reaction channel during the synthesis of iron oxide nanoparticles by thermolysis of $\text{Fe}(\text{acac})_3$ in 2-pyrrolidone, the brown deposits showing serious signs of fouling.

this also increased drag on the droplets, slowing them down and causing coalescence and irregularity in the droplet size.

Whatever the explanation for the contact of the droplets with the reaction channel walls, it became clear that it would not be straightforward to adapt the chosen synthetic route to the flow system we had developed. On the assumption that the problems we encountered were mostly due to the reaction solvent, similar syntheses in an alternative solvent were investigated.

5.3 Synthesis of Iron Oxide Nanoparticles in Phenyl Ether

Previous work on the high temperature synthesis of nanoparticles in flow reactors has been carried out within the group, and reported by Krishnadasan *et al*^{21,77} and Nightingale *et al*^{6,22,24,57}. Much of this work focused on the synthesis of chalcogenide quantum dots. The three-phase reactor developed by Nightingale *et al*⁵⁷ had been used for this purpose. In this case, the reaction solvent was octadecene (ODE) and it had been possible to achieve stable, controllable droplet flows even at high temperature. ODE was therefore a natural candidate for improving the flow characteristics of the current synthesis.

Unfortunately, $\text{Fe}(\text{acac})_3$ is insoluble in ODE and it was not possible to use a mixture of 2-pyrrolidone and ODE (to achieve good solubility of $\text{Fe}(\text{acac})_3$ and favourable flow characteristics) as 2-pyrrolidone is not miscible with ODE. Therefore a solvent needed to

be found that could either fully replace 2-pyrrolidone to give good droplet flow characteristics, or which was both miscible with ODE and a good solvent for $\text{Fe}(\text{acac})_3$.

A controlled flask-based synthesis of iron oxide nanoparticles by the thermal decomposition of $\text{Fe}(\text{acac})_3$ was reported by Sun *et al*¹⁰³, using phenyl ether as the reaction solvent. One disadvantage over the 2-pyrrolidone route was that it required a long-chain fatty acid, oleic acid, and a long-chain amine, oleyl amine, as the capping ligands. As well as being more complicated to implement than a synthetic route using the reaction solvent as the nanoparticle capping ligand, the ligands involved meant that the nanoparticles would not be soluble in water as-synthesised, requiring a further ligand exchange step afterwards. This was, however, seen as an acceptable compromise if it meant that a stable droplet flow could be achieved and reactor fouling could be avoided. A simplified version of this synthesis, reported by Xu *et al*¹⁵³, was adapted to a droplet flow reactor in the PhD thesis of Dr. Kumar¹⁴³. However, only very small (<4 nm) nanoparticles were obtained, and at very long (1 hr) residence times. The reaction time used by Sun *et al*¹⁰³ was only 30 min, and they were able to produce nanoparticles with a diameter of up to 16 nm. It was therefore hoped that adapting the synthetic route used by Sun *et al*¹⁰³ to flow would allow shorter residence times to be used to obtain larger nanoparticles.

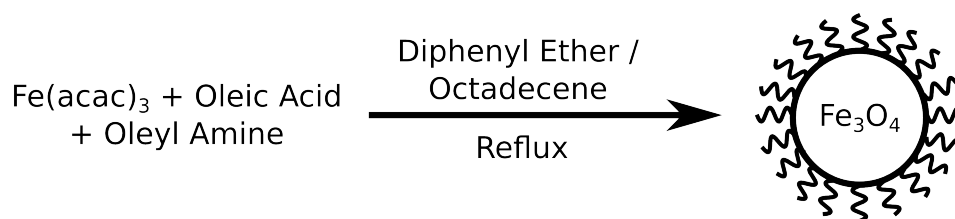


Figure 5.10: Synthesis of iron oxide nanoparticles by thermal decomposition of $\text{Fe}(\text{acac})_3$ in a mixture of phenyl ether and ODE, using oleic acid and oleyl amine as capping ligands.

Adapted from the synthesis reported by Sun and Zeng¹⁰³.

Initial tests with just phenyl ether showed that it behaved in a similar manner to 2-pyrrolidone in droplet-flow at high temperatures - the droplets would migrate to one side of the channel walls when they reached reaction temperature, risking fouling in a real synthesis. However, unlike 2-pyrrolidone, phenyl ether is miscible with ODE. A reaction mixture was adapted from the synthesis reported by Sun and Zeng, consisting of 353 mg (1 mmol) of $\text{Fe}(\text{acac})_3$, 802 mg of oleylamine (1.3 mL, 3 mmol), and 1.34 g of oleic acid (1.5 mL, 3 mmol) in 20 mL of a 50/50 (w/w) mixture of phenyl ether and ODE. The alcohol that Sun and Zeng used in their synthesis as a reducing agent, 1,2-hexanediol, was omitted because on a first attempt it formed a precipitate in the reagent syringe, most likely due to reduced solubility in the presence of ODE. Investigations into the role of the reducing agent in this synthesis reported in the literature suggest that it is not necessary for formation of magnetic iron oxide nanoparticles¹⁵⁴. The reaction scheme of the synthesis used is shown in Fig. 5.10

5.3.1 A Working Three-Phase Synthesis of Iron Oxide Nanoparticles

Using this reaction mixture in the same three-phase flow reactor as before gave promising results. After systematically varying the reactor conditions, a stable droplet flow was achieved at 240 °C using $Q_{\text{gas}} = 0.4 \text{ scc/min}$, $Q_{\text{carrier}} = 100 \text{ }\mu\text{Lmin}^{-1}$, and $Q_{\text{reagent}} = 50 \text{ }\mu\text{Lmin}^{-1}$ with a back-pressure of 5 psi. After allowing the droplet flow to properly stabilise over 15 min, a video was taken. The first frame of this video can be seen in Fig. 5.12a. For the duration of the video there was no observable change in the colour of the reagent droplets along the flow profile. The droplets retained the bright red colouration of the reaction mixture, suggesting no iron oxide nanoparticles were formed.

To confirm this, the colour of a section of the reaction channel at the reactor inlet and of a section at the reactor outlet was tracked through each frame of the video. For this colour analysis, the Red Green Blue (RGB) values of a 4×4 grid of pixels at the centre of each circle were measured for every frame of the video. Each RGB value is composed of three numbers, defining how much red, blue, and green are present in a particular colour. This makes comparing RGB values to each other difficult, as they do not define a linear progression from one colour to another. To better compare the colours of the droplets, and see if any reaction had occurred at all, the RGB values were converted from the cubic RGB space to a cylindrical Hue Saturation Value (HSV) colour space (see Fig. 5.11a and 5.11b). The hue in HSV colour space defines a progression from one colour to another (see Fig. 5.11c), and therefore allowed changes in the colour of the droplets between the two positions in the reaction channel to be more easily observed.

For this analysis, a script was written in the Python programming language to load the frames of the video and calculate the hue from the RGB values of the selected pixels. The hue was calculated using the following set of equations:

$$C_{\text{max}} = \max(R, G, B) \quad (5.1a)$$

$$C_{\text{min}} = \min(R, G, B) \quad (5.1b)$$

$$C = C_{\text{max}} - C_{\text{min}} \quad (5.1c)$$

$$H' = \begin{cases} 0 & \text{if } C = 0 \\ \frac{G-B}{C} \bmod 6 & \text{if } C_{\text{max}} = R \\ \frac{B-R}{C} + 2 & \text{if } C_{\text{max}} = G \\ \frac{R-G}{C} + 4 & \text{if } C_{\text{max}} = B \end{cases} \quad (5.1d)$$

$$H = 60^\circ \times H' \quad (5.1e)$$

where R, G, and B were the values of red, blue, and green in the RGB values, C_{max} was the maximum value for a set of RGB values, and C_{min} was the minimum. These were used to calculate the chroma value, C , and the fractional hue, H' . The fractional hue was then converted to the hue, H , which had units of degrees.

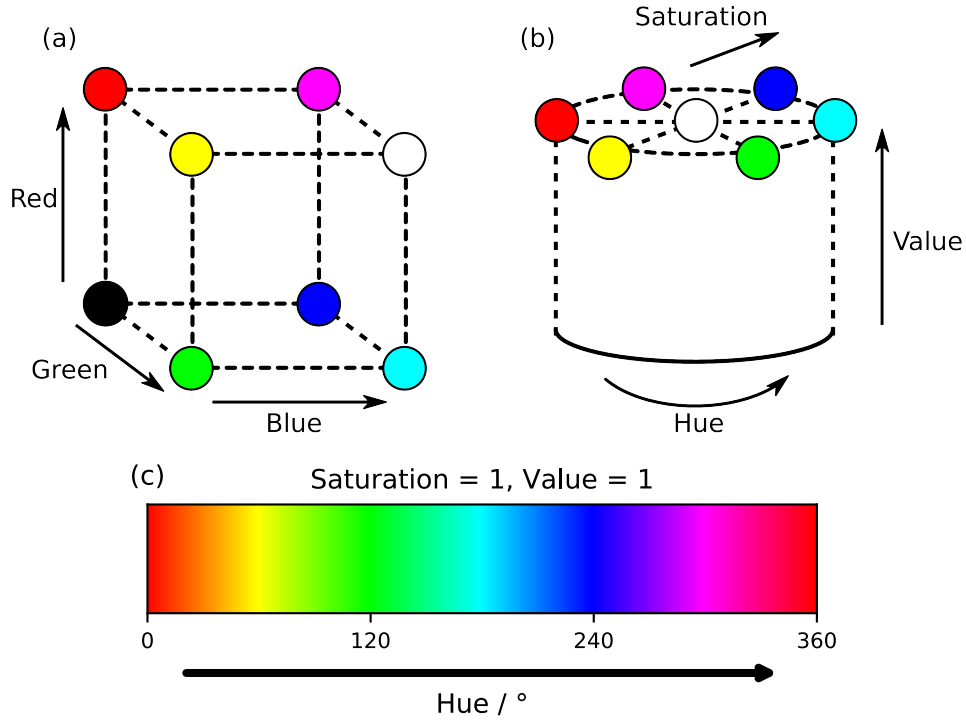


Figure 5.11: Diagrams of (a) the cubic shape of the RGB colour space, (b) the cylindrical shape of the HSV colour space, and (c) the change in colour with increasing hue value, at maximum saturation and value.

The sections that were analysed are shown in Fig. 5.12a, with the inlet position circled in red, while the outlet has been circled in black. The hues were averaged across these 4×4 grids of pixels, giving a single value at each position that was tracked over time, as shown in Fig. 5.12b.

Both the inlet and the outlet had a periodic increase then decrease in the average hue of the pixels analysed. This can be attributed to the passage of droplets through the positions chosen at the inlet and outlet of the reactor. In the HSV colour space used here, pure red had a hue of 360° , pure blue had a hue of 240° , and pure green had a hue of 120° . As such, the periodic peaks in Fig. 5.12b represent a red colour, and therefore the presence of a droplet of the red reaction mixture at the chosen position in the video frame. The periodic drop to below 240° may seem puzzling, as the reaction channel without any droplets appears white in Fig. 5.12a. However, in the HSV colour

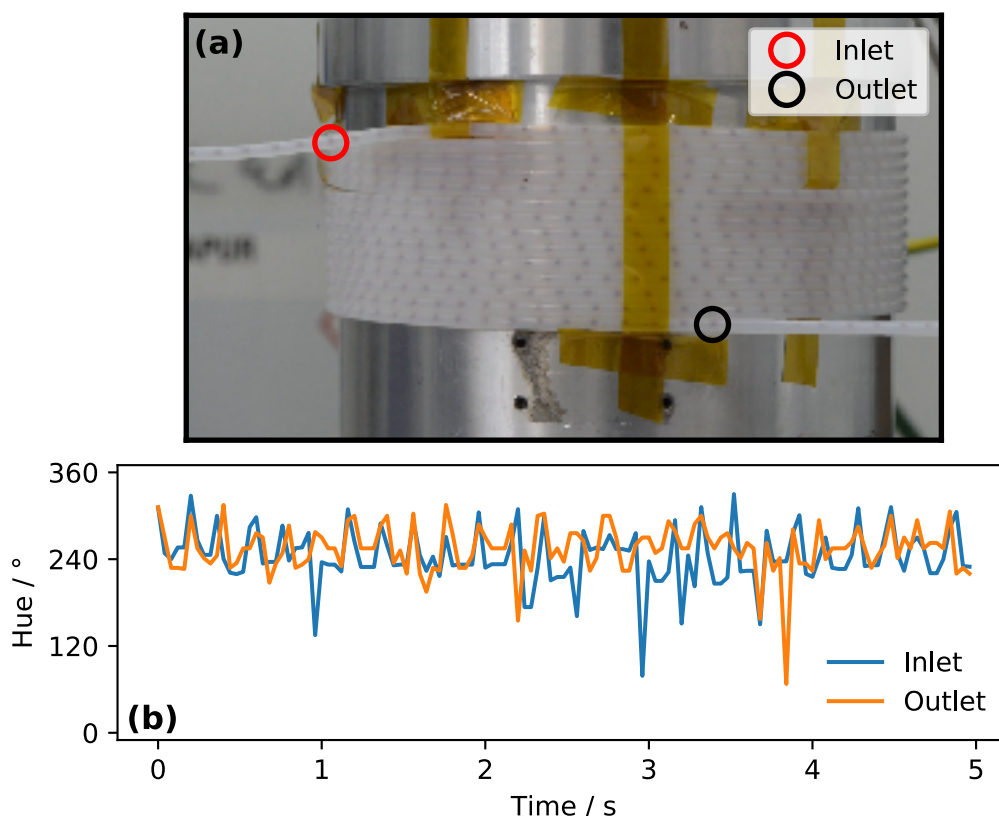


Figure 5.12: (a) A frame from the video taken of the reactor running at $240\text{ }^{\circ}\text{C}$, $Q_{\text{gas}} = 0.4\text{ scc/min}$, $Q_{\text{carrier}} = 100\text{ }\mu\text{Lmin}^{-1}$, and $Q_{\text{reagent}} = 50\text{ }\mu\text{Lmin}^{-1}$ with a back-pressure of 5 psi, indicating the positions used for droplet colour analysis at the inlet (red) and outlet (black), and (b) a graph of the hue at these positions over the first 5 seconds of the video.

space white and black can be formed at all hue values, by having a saturation of 0 and a value of 1 or 0 respectively. The colour of the reaction channel corresponded to the minima of the hue values shown in Fig. 5.12. The inlet of the reaction channel had an average hue of 215 ° and the outlet had an average hue of 208 ° . As such, the reaction channel showed as blue in the video, but with a very low saturation.

There was a large overlap of the average hue at the inlet and outlet positions, both in terms of hue and the time at which the peaks occurred. The overlap in time suggested that the frequency of the droplets, and therefore the velocity, was constant throughout the reactor. This confirmed that a stable flow had been attained, unlike in previous experiments where the spacing between droplets reduced along the length of the reaction channel. The close overlap in hue at the peaks in Fig. 5.12b suggested that the colour of the droplets at the inlet and outlet was the same, and that no reaction had occurred under the conditions employed.

To confirm this, the hues at the peaks in Fig. 5.12b were plotted against time, as shown in Fig. 5.13a. A simple method was used to find the peak at each local maximum, by finding all points with a hue above a threshold value, 290 ° in this case, and then finding the

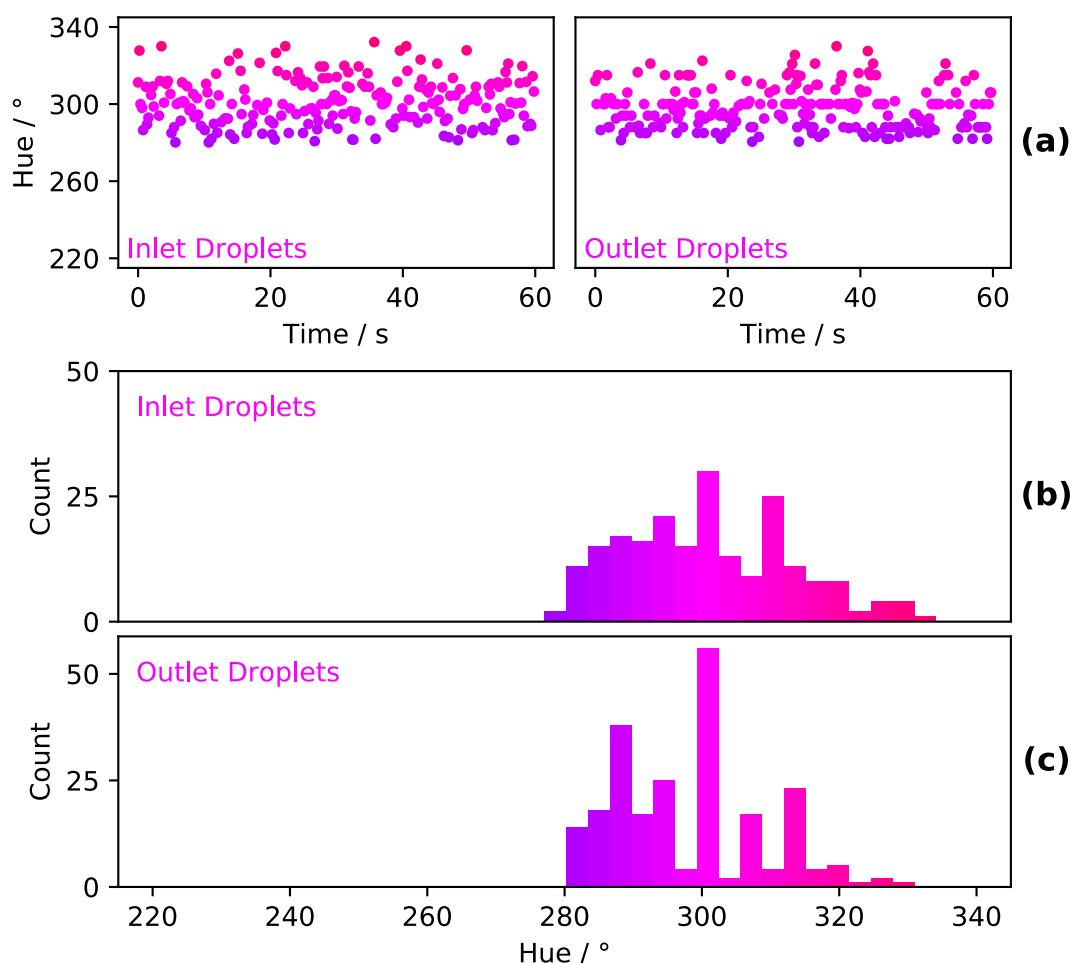


Figure 5.13: The hue of the droplets at the inlet and outlet positions over the course of the video (a) with the points coloured by their hue, and the distribution of droplet hues at (b) the inlet position and (c) the outlet position.

maximum for each set of consecutive points. Although this may miss some peaks, enough were identified for the data to be representative of the colour of the droplets at the inlet and outlet positions. If iron oxide nanoparticles had formed, the droplets at the outlet would have been a much darker purple-black colour, as observed during nanoparticle formation in batch. In terms of Fig. 5.13a, this would be expected to manifest as a separation in the hues of the droplets, with the droplets at the inlet having higher hue values than those at the outlet. There was a slight separation between the hues of the droplets at the inlet and the outlet. Each point in Fig. 5.13a has been coloured using the hue of the droplet that it represents, at maximum saturation and value. This shows that although there was some variation in the colour of the droplets, they were all very similar. This confirmed that the colour of the droplets was similar at the inlet and outlet of the reactor, and therefore relatively few iron oxide nanoparticles were formed.

To more clearly see the distribution of colours of the droplets at the inlet and outlet positions, histograms were made of the average hues, shown for the inlet position in

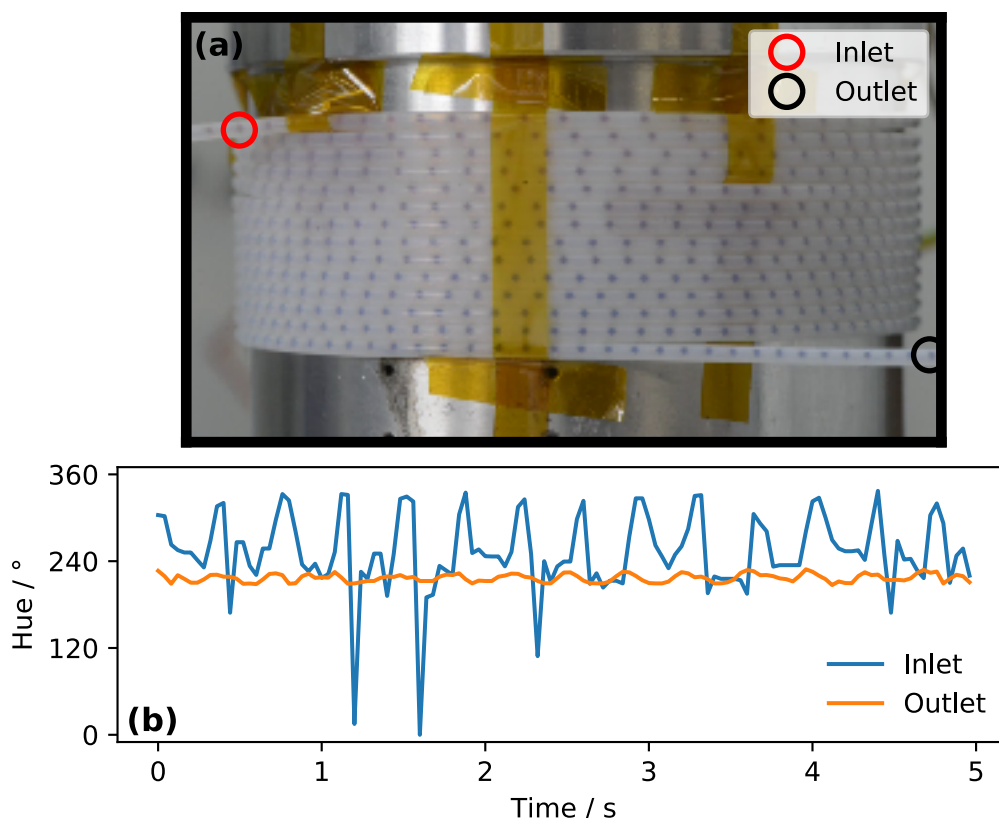


Figure 5.14: (a) A frame from the video taken of the reactor running at 240 °C, $Q_{\text{gas}} = 0.3$ scc/min, $Q_{\text{carrier}} = 80 \mu\text{Lmin}^{-1}$, and $Q_{\text{reagent}} = 40 \mu\text{Lmin}^{-1}$ with a back-pressure of 5 psi, indicating the positions used for droplet colour analysis at the inlet (red) and outlet (black), and (b) a graph of the hue at these positions over the first 5 seconds of the video.

Fig. 5.13b and for the outlet in Fig. 5.13c. The distributions appear to be very similar, with most of the droplets having a hue near 300 °. Once again, the bars have been coloured according to the RGB representation of their hue at maximum saturation and value, showing that the most common droplet colour was magenta. The mean hue for the distribution of droplets at the inlet position was 301 ° with a standard deviation of 12 °, compared to a mean hue of 298 ° for the outlet droplet distribution and a standard deviation of 11 °. From this it is clear that there was no formation of iron oxide NPs under these conditions.

Having failed to achieve nanoparticle formation at the selected reaction conditions, alternative conditions were tested to see if any iron oxide nanoparticles could be made. It was thought that longer residence times might lead to particle formation, so slower droplet flows were attempted. As the residence time could not be directly controlled, the gas and liquid flow rates were reduced until the colour of the droplets darkened sufficiently along the length of the reactor to suggest significant formation of iron oxide nanoparticles. This was achieved at 240 °C, $Q_{\text{gas}} = 0.3$ scc/min, $Q_{\text{carrier}} = 80 \mu\text{Lmin}^{-1}$, and $Q_{\text{reagent}} = 40 \mu\text{Lmin}^{-1}$, with 5 psi of back-pressure. Once again, a video was taken after the droplet-flow

of the reaction had been stable for at least 15 min, shown in Fig. 5.14a.

To confirm that iron oxide nanoparticles were being formed at these conditions, the colour of a 4×4 grid of pixels was analysed again at the reactor inlet (circled in red) and the reactor outlet (circled in black). Carrying out the same colourimetric analysis as before yielded the graph shown in Fig. 5.14b, showing the change in average hue at the two positions over time. A similar picture as before can be seen, with the hue periodically rising and falling as droplets passed into and out of the selected sections of the reaction channel. However, the peak maxima of the hues at the outlet position were much lower than those at the inlet, with the former remaining below 240 °.

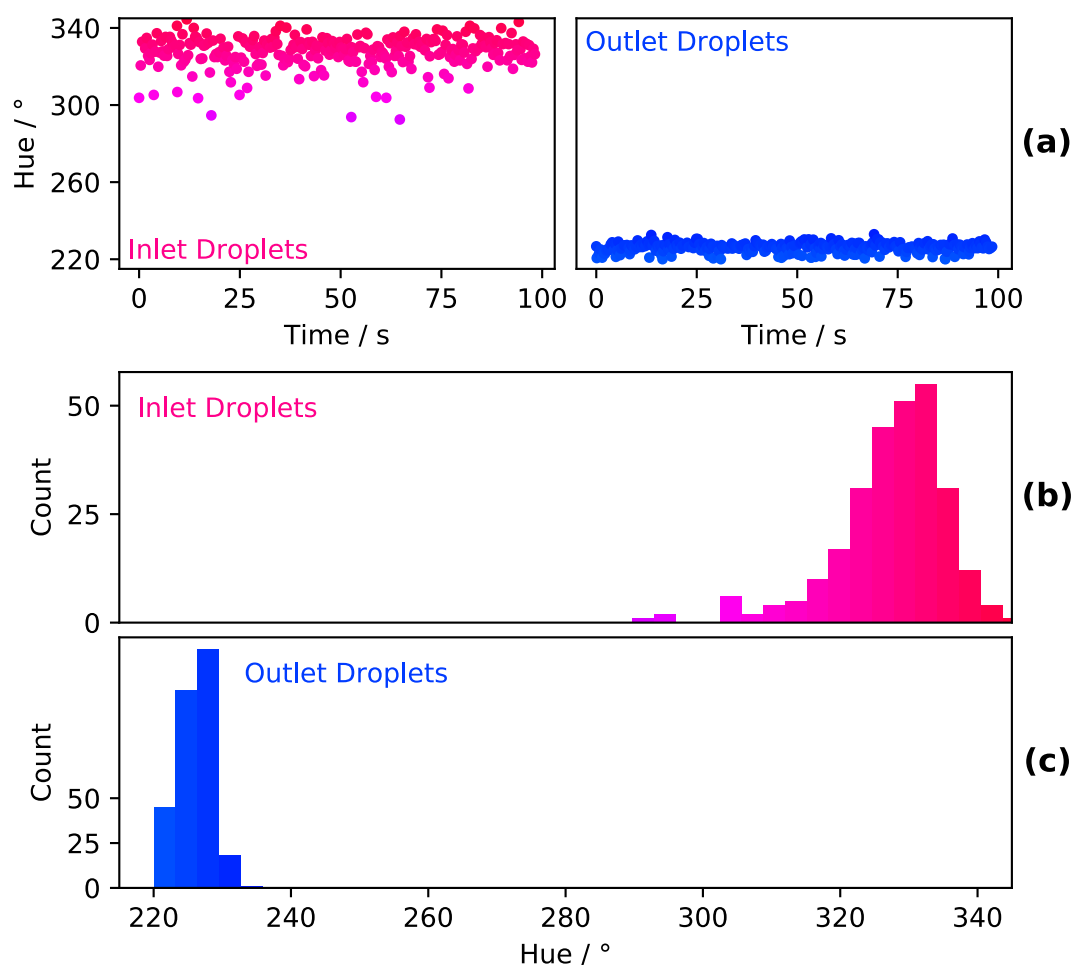


Figure 5.15: The hue of the droplets at the inlet and outlet positions over the course of the video (a) with the points coloured by their hue, and the distribution of droplet hues at (b) the inlet position and (c) the outlet position.

The hue at the peak maxima was found using the same method as before, this time using a threshold of 290 ° for the inlet position and 220 ° for the outlet. As can be seen from Fig. 5.15a, there was a much more distinct difference between the hue of the droplets at the inlet and the outlet than at the previous set of conditions. Colouring the points with their hue, at maximum saturation and value, shows that the droplets at the outlet

were much more blue and the droplets at the inlet were much closer to red, suggesting a change in colour consistent with the formation of iron oxide nanoparticles.

The distributions of droplet hue are displayed in Figs. 5.15b and 5.15c. This further confirms that there were significant differences in colour between the droplets at the reactor inlet and at the reactor outlet, with two skewed distributions of hue that do not overlap. The mean droplet hue at the inlet was 328° with a standard deviation of 8° , while the mean droplet hue at the outlet was 226° with a standard deviation of 2° . The large difference in hues confirmed that the colour of the droplets changed at these reactor conditions, from the red of the reagent mixture to a much darker black-purple colour indicative of the formation of iron oxide nanoparticles.

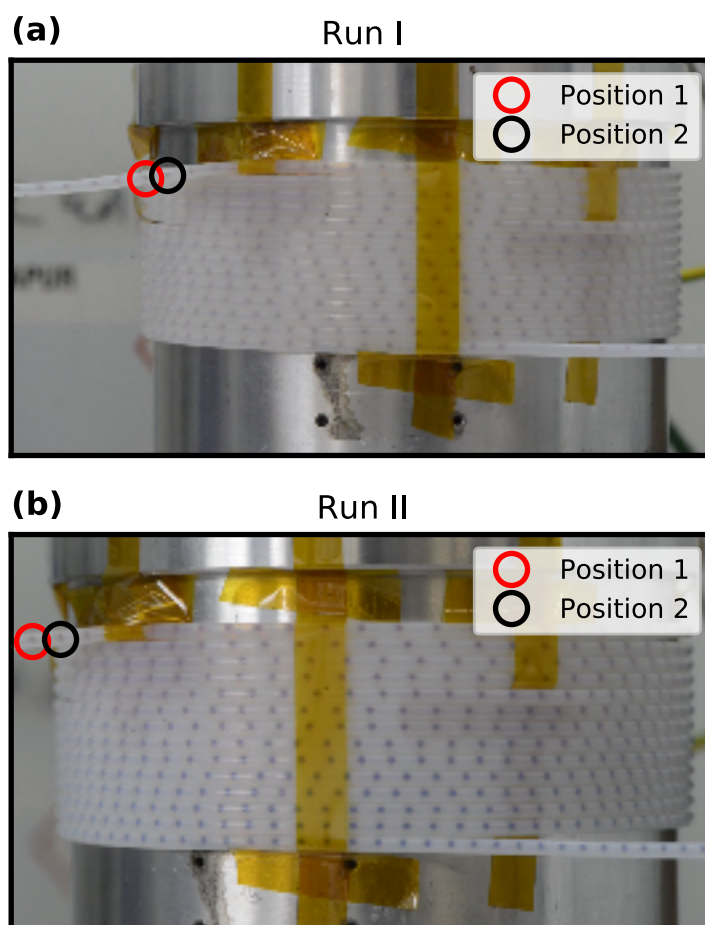


Figure 5.16: The first frame of the videos taken of (a) Run I and (b) Run II, indicating the two positions used to determine the droplet velocity.

As the reactor used a single reagent inlet, the concentration of reagents in the reaction mixture was fixed in these experiments. Both syntheses shown here were carried out at the same temperature, so the main change that could have made the difference between having no reaction and successful nanoparticle production was the residence time. In a continuous or two-phase droplet flow system, the residence time could be estimated by the total liquid flow rate and the volume of the reaction channel that the reaction occurred in

(the heated volume in this case). However, this is not as easy in a three-phase system. As well as the difficulty of converting between the different flow rate units, the gas in a three-phase system is compressible. The volume of the gas in the reactor may therefore not be the same as the volume of gas pumped into the reactor, and the relationship between the two may be non-linear. This is worsened by the use of applied back-pressure, which serves to actively compress the gas in the system.

While droplet tracking software has been used in previous three-phase systems to estimate the droplet velocity, and therefore the residence time, a simpler method was used in this case. Similar to how the colour of the droplets was estimated, the average hue over two 4×4 grids of pixels was tracked during the course of the videos of the two syntheses. This time, both positions were at the inlet of the reactor, chosen as the position of two consecutive droplets in the first frame of each video. These positions can be seen in Fig. 5.16a for Run I, and Fig. 5.16a for Run II.

The change in hue of these positions over time are shown for Run I and II in Fig. 5.17a and Fig. 5.17b respectively. The periodic rise and fall in hue as the droplets passed through the positions in the reaction channel can once again be seen. What is important in this case, however, is not the peak maxima, but that the timing of the peaks overlapped almost exactly. This means that the frequency of droplets moving through both positions is the same, suggesting that for a particular peak in the hue at Position 1, the droplet that it corresponds to shows up as the following peak in the hue at Position 2. Knowing the distance between the two positions, this allowed the velocity of the droplets to be calculated by first finding the frequency of the droplets, converting that to the average time between droplets at a particular position, and then dividing the distance between the two positions by that time.

Although it may have been simpler to calculate the difference in time between the peaks in the hue at Position 1 and the following peak in hue at Position 2, the noise in the data meant that some peaks were likely missed, and some noise misidentified as peaks. A bigger criticism of this method for estimating the average droplet velocity is that the droplets might have been moving faster than the frame rate of the video, so a peak in the hue at Position 1 did not necessarily correspond to the same droplet as the following peak in the hue at Position 2. However, at a frame rate of 25 frames per second, this would have occurred at much faster flow rates.

The frequency of the droplets at Position 2 was estimated by converting the change in hue over time to the frequency domain, using a fast fourier transform. The resulting frequency spectra are shown in Figs. 5.18a and 5.18b for Run I and Run II respectively. Sharp peaks around 5.0 Hz in Run I and 2.5 Hz for Run II can be seen, showing that during both syntheses there was a stable droplet-flow by the relatively narrow distribution of frequencies. In the frequency spectrum for Run II, there was also a smaller, broader peak at just above 5 Hz, which can be attributed to the second harmonic of the droplet

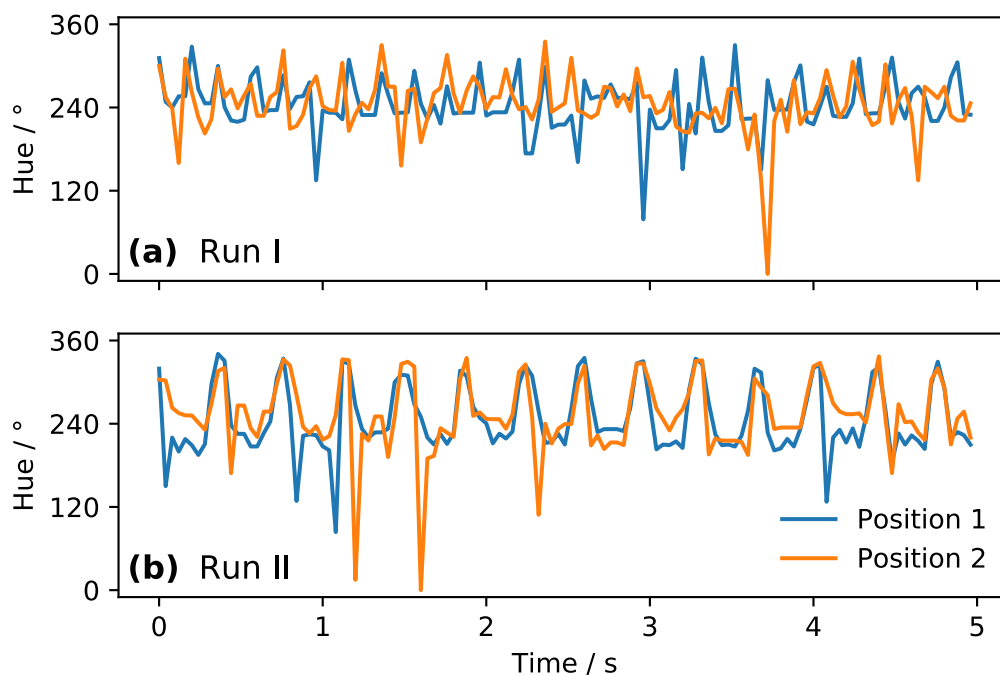


Figure 5.17: The hue at the two inlet positions for the first 5 seconds of the videos of (a) Run I and (b) Run II.

frequency. Given the noise in the signal for both runs, the mean and standard deviation of the droplet frequencies were found by fitting a Gaussian, using gradient descent to minimise the mean squared error of the fit.

Both Gaussian fits had a MSE below 0.20, indicating a relatively good fit, with the slightly high error rate most likely due to the amount of noise in the signals. For Run I, the mean droplet frequency from the curve fitting was 5.12 Hz and the standard deviation was 0.054 Hz. Using twice the standard deviation as an estimate for the error within a 95 % confidence interval, the time between droplets for Run I was 0.195 s (± 0.004 s). For Run I, the distance between Position 1 and Position 2, i.e. the distance between droplets, was 2.53 mm, resulting in an estimated velocity of 0.50 mm/s. In Run II, the mean droplet frequency was 2.77 Hz, and the standard deviation was 0.072. This translated to a time of 0.361 s (± 0.019 s), or an estimated droplet velocity of 1.46 mm/s (for a distance of 1.90 mm between Position 1 and 2). The same reactor set up was used for both experiments, so the heated length of the reaction channel was 3.52 m for both. This means that the estimated residence time for Run I was 4.5 min compared to 8.7 min for Run II, demonstrating that by doubling the residence time we were able to successfully form iron oxide nanoparticles. A summary of the results for these two syntheses can be seen in Table 5.1.

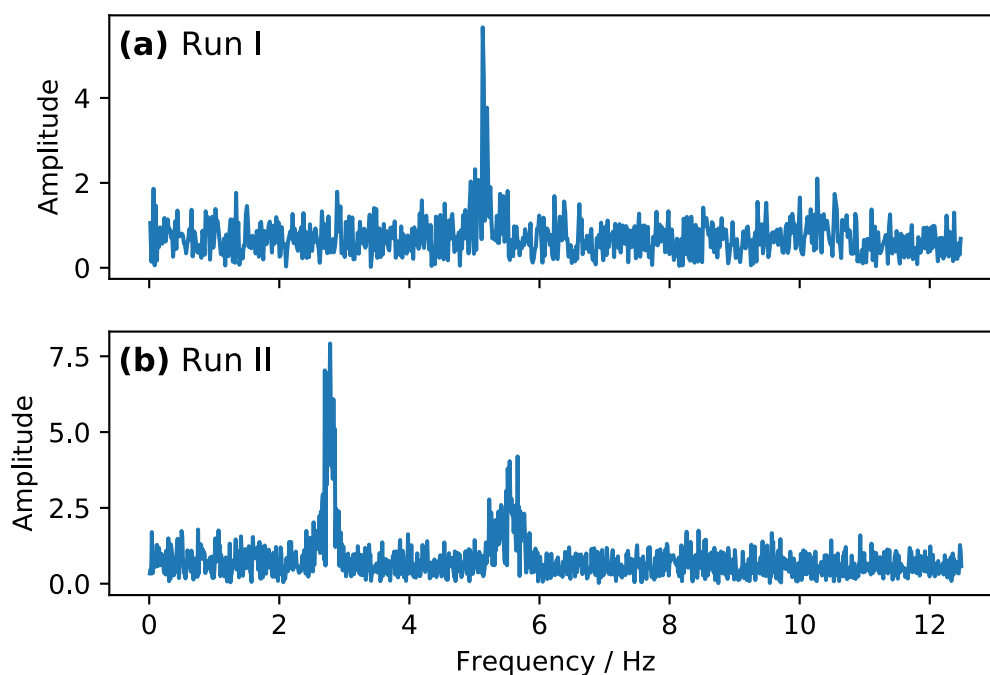


Figure 5.18: The frequency spectrum of the the droplet hues at the two inlet positions for (a) Run I and (b) Run II.

5.4 Conclusion

The work in this chapter focused on adapting the synthesis of iron oxide nanoparticles, via the thermal decomposition of $\text{Fe}(\text{acac})_3$, to a flow system with a view to exerting control over the size, and therefore magnetic properties, of the particles. Building on work by Dr. Thomas Phillips, a simple synthesis involving 2-pyrrolidone as both the reaction solvent and nanoparticle capping ligand was tried. A custom designed multi-phase back-pressure regulator, built by Dr. James Bannock, was used to gain a stable droplet-flow, without any pulsing of the gas flow as had been encountered previously.

	Residence Time / min	Hue / °	
		Inlet	Outlet
Run I	4.5	301 (± 12)	298 (± 11)
Run II	8.7	328 (± 8)	226 (± 2)

Table 5.1: Summary of results from Run I (240 °C, $Q_{\text{gas}} = 0.4$ scc/min, $Q_{\text{carrier}} = 100$ μLmin^{-1} , and $Q_{\text{reagent}} = 50$ μLmin^{-1} , 5 psi) and Run II (240 °C, $Q_{\text{gas}} = 0.3$ scc/min, $Q_{\text{carrier}} = 80$ μLmin^{-1} , and $Q_{\text{reagent}} = 40$ μLmin^{-1} , 5 psi) showing the estimated residence time and droplet hue at reactor inlet and outlet for both.

A reproducible method for attaining a stable droplet-flow in the system was developed, and used to investigate the minimum temperature at which iron oxide NPs would form in the reactor. It was found that nanoparticles were only formed at the maximum safe temperature of the flow reactor, 240 °C. However, even with a stable droplet-flow going in to the reactor, droplet coalescence still occurred, leading to reactor fouling.

An alternative synthesis was then tried using a mixture of phenyl ether and ODE as the reaction solvent. This combined the solubility of Fe(acac)₃ in phenyl ether with the good flow characteristics of ODE that had been previously reported in the literature. A stable gas-liquid flow was once again attained, and control was demonstrated over the residence time of the synthesis with no fouling. Finally, iron oxide NP formation was observed at higher residence times, showing that the goal of successfully adapting the high temperature synthesis of iron oxide nanoparticles to a flow system was achieved, and the reactor fouling that had been a problem before had been prevented.

5.4.1 Further Work

While the synthesis of iron oxide nanoparticles in flow was shown here, and some degree of control over the residence time was demonstrated, results reported here are only preliminary. In particular, control over the size of the iron oxide nanoparticles was not demonstrated. However, the work described here provides a good starting point for systematic investigations into the effects of reaction conditions, such as residence time and reagent ratios, on the size of the particles produced. The use of a three-phase flow reactor also opens up the possibility of carrying out multiple additions of reagent mixture to the droplets, and thereby producing larger nanoparticles or particles of radially varying composition.

In order to see the effects of varying the reaction conditions, the iron oxide nanoparticles need to be characterised both in terms of particle size and magnetic properties. Microscopy techniques such as SEM or TEM are necessary for measuring the particle size, while magnetic properties can be measured by vibrating sample magnetometry (VSM). Neither of these have been adapted for in-line analysis on a flow reactor, and neither could be easily adapted.

The method used here to determine if the reaction was successful by measuring the colour of the reagent droplets could be easily adapted to give real time information about the system. The use of this method was based on characterisation of a three-phase droplet flow reported by Nightingale *et al*⁵⁷, and a real time in-line colorimetric droplet analysis might look more like the set-up they used - controlling the lighting of the reaction channel with light boxes to give well-defined droplets, and recording with a computer-linked microscope. As well as giving a measure for how much iron oxide has been formed, this would also allow the droplet residence time to be calculated from real-time information

about droplet velocity, and therefore give direct control over the residence time of the synthesis.

Finally, although adaptation of electron microscopy techniques to in-line analysis of flow reactors is infeasible, developing in-line measurement of magnetic properties using techniques other than VSM might be a more achievable goal. This would allow for faster analysis of the iron oxide nanoparticles being produced, and allow for automatic optimisation of these magnetic nanoparticles.

CHAPTER 6

Conclusions and Further Work

6.1 Conclusions

This thesis has described work on the advancement of flow reactor technology for the controlled synthesis of high-quality functional materials, focusing on two specific areas. The first area was the development of a more intuitive method for optimising competing requirements of a chemical reaction, which also involved constructing an automated flow reactor for an example synthesis. The second area was the application of a recently reported three-phase flow reactor to the synthesis of magnetic iron oxide nanoparticles, adapting the reactor design to the challenges of this particular reaction.

The synthesis of *o*-xylenyl functionalised fullerene derivatives was chosen as an example of a reaction with competing requirements that needed to be optimised. The cascadic nature of this synthesis means that when a particular product forms, it can then be consumed to form another, resulting in a mixture of products. However, only one of the products is useful, while the others are less useful or actively detrimental to their applications, meaning the trade-off between achieving a high yield of the useful product and minimising the amount of other products needed to be optimised.

An automated reactor was developed to carry out this synthesis in flow. This was done to take advantage of the more uniform and more controllable reaction conditions compared to conventional batch chemistry, as well as the fact that the low volumes typical of flow reactors saved on time and materials when carrying out larger experiments, such as optimisations.

A self-built flow reactor was constructed to provide flexibility in the configuration, and in the methods used to control it. The reactor was composed of a dual-channel syringe pump that delivered the two reagent solutions to the small-diameter tubing reaction channel, through a y-shaped mixer. The reaction channel was wound around a self-built solid-state heater, and the exit of the reaction channel was connected to the injection valve of a self-built high performance liquid chromatography (HPLC) system, to allow real-time analysis of the product mixture.

Experiments were carried out to investigate the stability and reproducibility of the synthesis of *o*-xylenyl functionalised fullerenes in this automated flow reactor. By setting it to run a synthesis at the same reaction conditions every 3 hours for 51 hours, it was demonstrated to give very good stability, with the mole fractions of all of the adducts in the product mixtures having a standard deviation of less than 1 %. The reproducibility of the synthesis was tested by synthesising product at the same reaction conditions with five different batches of the sultine starting material - as this was identified as the largest possible source of variation in the synthesis. While this showed that there was increased variation in the product mixture when using different batches of sultine stock solution, the variation was still very low.

With the reproducibility and stability of the reactor demonstrated, a large automated

experiment was carried out. This involved mapping the parameter space of the reaction by sampling points from it that formed a $6 \times 6 \times 6$ grid. Replicates of some of the points were carried out periodically, resulting in an experiment that synthesised product at 246 reaction conditions over the course of a few days. This was a clear demonstration of the advantage of an automated flow reactor; such a large experiment would have been very labour intensive had it been carried out in batch, but in this case only needed a chemist to make the stock solutions and provide a list of reaction conditions to the reactor control function.

Based on the results of this parameter space mapping experiment, automated optimisations were carried out that attempted to optimise a trade-off between maximising the desirable product, X_2 , while suppressing formation of the undesirable product, X_3 . These experiments also demonstrated the difficulty of formulating a merit function using the weighted-sum approach, as none of the optima found gave the desired balance of high yields of the desirable product with none of the undesirable product. Similarly, when an optimisation with a merit function designed solely to maximise the yield of the most desirable product was carried out, the optimum identified had very high levels of the undesirable product as well.

As a more intuitive alternative to using a weighted combination of the product mole fractions, a method proposed by Huyer and Neumaier⁹⁵ was used to encode the requirements of the synthesis. This involved formulating the problem as a constrained optimisation, where a lead property was chosen as the objective function to minimise, and any other requirements were formulated as constraints with upper and lower limits. The merit function was then composed of two parts: a scaling of the objective function by the value of some previously identified point that satisfies all of the constraints, and a penalty imposed on any violations of the constraints.

A series of optimisations were carried out with different values of a parameter that defined how strict the penalty for constraint violation was, to gain an understanding of how the SNOBfit optimisation algorithm behaved when used with this constrained optimisation method. With this better understanding, this method was then used to optimise the synthesis to a particular product specification. Through trialling a few different specifications, a specification was found gave the best balance of X_2 and X_1 , with X_3 as less than 10 % of the product. In doing so, the ease with which a product specification could be encoded into a constrained optimisation was demonstrated, identifying this as an intuitive and powerful method for optimising chemical systems with competing requirements.

The final chapter of this thesis focused on the development of a different area of flow chemistry: the application of three-phase reactor technology to new synthetic challenges. Using the method of three-phase flow synthesis reported recently by Nightingale *et al* for the multi-step synthesis of quantum dot nanoparticles, a flow reactor was built for the controlled synthesis of magnetic iron oxide nanoparticles by the thermal decomposition

of $\text{Fe}(\text{acac})_3$. The main challenges in this synthesis were obtaining a stable gas-liquid flow and preventing fouling of the reactor. The former was achieved through use of a bespoke back-pressure regulator and careful operation of the reactor. The second was achieved by using a modified synthetic procedure, where half of the reaction solvent was substituted for ODE to give better flow characteristics. Video analysis of the droplets from two syntheses then showed that an increase in the residence time from 4.5 min to 8.7 min was enough to go from no nanoparticles to nanoparticle formation. Therefore, control over the synthesis was demonstrated through control of the average residence time.

6.2 Further Work

6.2.1 Automatic Optimisation of More Challenging Targets

The work in the first two chapters of this thesis described the construction of an automated reactor, and its use for the development of a more intuitive way of encoding a chemist’s requirements in a chemical optimisation, through the use of constrained optimisations. While this was successfully applied to the cascadic synthesis of *o*-xylenyl functionalised fullerene derivatives, the chemistry of this reaction is well known, and so the results were not unsurprising.

To further demonstrate the power of this method, and widen its applications, it is necessary to use these constrained optimisations to optimise syntheses with more challenging requirements. This could be a synthesis where the interaction of requirements is less clear, for example the synthesis of magnetic iron oxide nanoparticles described in the final chapter. Different applications of the magnetic nanoparticles require different average diameters, but as large a magnetic susceptibility as possible. However, the size is non-trivially linked to the magnitude of magnetism of the nanoparticles, with it also depending on the quality (amount of defects) of nanoparticles as well. Therefore, it is not obvious what the optimal trade-off between magnetic susceptibility and nanoparticle size is, especially given the requirements of different applications. Similarly, a synthesis with a large number of reaction parameters that can be varied would provide another challenge, such as shape-selective nanoparticle syntheses where the amount and type of capping ligands, as well as the reaction solvent, reagent ratios, and temperature can have a profound effect on the size, shape, and quality of the nanoparticles - making tunable syntheses more difficult using batch chemistry.

Neither of these cases have been examined in the literature, with most examples of self-optimising flow reactors focusing on low dimensional problems, with at most two competing requirements. The methods described here would be especially powerful in the case of a synthesis with a large number of competing requirements, as encoding those in a set of constraints would be a simpler process than formulating a weighted-sum or weighted-

product figure of merit. Further, the tools developed for running constrained optimisations with SNOBFit in this thesis are compatible with any flow reactor configuration, and so could be applied to reactors with different hardware, or different methods of in-line analysis.

6.2.2 Handling Uncertainty

The idea of using constrained optimisations in this thesis was to make the process of carrying out a chemical optimisation more intuitive, and informative, for a chemist. One area that has not received much attention in the field of self-optimising flow reactors is how they handle uncertainty. By their nature, optimisation algorithms do not sample the whole parameter space. As such, there will be an amount of uncertainty about whether or not parts of the parameter space that have not been evaluated contain more a more optimum point than the one already identified.

Typically, an optimisation is terminated when either the currently identified optimum point has not changed for a certain number of function evaluations, when the value at the optimum point is below a particular threshold, or when the optimisation has performed a set number of total function evaluations. While all of these tactics may give a high likelihood that the optimum, or at least an optimum, has been found, for black-box systems like chemical optimisations it is difficult to determine if the optimisation has been terminated too early. By using an optimisation routine that could return information about the uncertainty in the value of the function at all points that have been evaluated, the chemist would be more empowered to make judgements about how well an optimisation has performed.

Optimisation routines that model the underlying function of the system are possibly the most appropriate for handling this uncertainty. This is because the uncertainty in the model can be used as an estimate of the uncertainty in parts of the parameter space that have not been evaluated. While the algorithm used in the work described in this thesis, the SNOBFit algorithm, does fit models to the data from previous function evaluations, these are only local fits to the points with the lowest objective function values. As such, this does not provide any information about the uncertainty in the unexplored parts of the parameter space.

One method that may offer a way of solving this problem is Bayesian Optimisation, which is already a well established method for optimising high-dimensional problems in machine learning, that are often expensive to evaluate¹⁵⁵. In a Bayesian Optimisation, the underlying function of the system is modelled by a prior distribution, that captures the users beliefs about the system. Each time the objective function is evaluated, the prior is updated with this information to give the posterior distribution, which reflects the current knowledge about the system. This posterior is used in the calculation of an

acquisition function, which selects the point with the highest likelihood of returning the optimum as the next point to evaluate.

The most commonly used prior in Bayesian Optimisations is a Gaussian process. This is the generalisation of a Gaussian distribution over functions. A Gaussian process is defined by a mean function and a covariance function. This makes the optimisation a process of finding the best mean function and covariance function to represent the system, where the mean function represents the modelled value of the system over the parameter space, and the covariance function gives the uncertainty in that value. Therefore, in a Bayesian Optimisation, as well as returning an estimated model of the system after each objective function evaluation, the uncertainty of the model is returned without any further calculations needing to be carried out.

Combining information about the uncertainty in the optimisation with the method developed here for optimising chemical systems with competing requirements would add a powerful tool to the toolbox of chemical optimisations. Rather than running an optimisation for a set number of function evaluations, an optimisation could be terminated when an optimum has been found with a desired likelihood - possibly allowing much fewer evaluations to be used. This could be especially useful in cases where a feasible point is difficult to find - by using the uncertainty of the modelled system, the chemist would know how likely it is that a feasible point exists somewhere else in the parameter space, if it does exist at all.

6.2.3 Iron Oxide

The work described in the final chapter of this thesis provides the groundwork for a controllable synthesis of magnetic iron oxide nanoparticles in flow, with alterations to a previously described three-phase flow reactor being made to better accommodate the challenges of this synthesis. While rudimentary control over the reaction through the residence time was demonstrated, further work is required to demonstrate size control of the nanoparticles, as well as characterise their magnetic properties. This would involve systematically varying the reaction conditions, such as residence time and gas and liquid flow rates. The design of the reactor could be expanded so that separate syringe pumps are used to mix the reagents in-line, before the reaction mixture is fed into the droplet reactor, to allow the ratios of capping ligands and iron salt to be varied as well. The size of iron oxide nanoparticles synthesised in this way would need to be measured using electron microscopy, and the magnetic properties could be measured using a technique like vibrating sample magnetometry. This would build a detailed understanding of the relationship between the reaction parameters and the size and magnetic properties of the iron oxide nanoparticles.

Appendix

Code for Control of the Automated Reactor Components

“pidBox.ino”

Arduino script loaded onto the solid-state heater microcontroller to control the temperature of the heater.

```
#include <Wire.h>
#include <Adafruit_MCP23017.h>
#include <Adafruit_RGBLCDShield.h>
#include <PID_v1.h>
#include <PID_AutoTune_v0.h>
#include "Adafruit_MAX31855.h"

#define RED 0x1
#define YELLOW 0x3
#define GREEN 0x2
#define TEAL 0x6
#define BLUE 0x4
#define VIOLET 0x5
#define WHITE 0x7
#define RelayPin 9

String msg;
String Status;
String tempstr;
int lastprnt;

double Setpoint, Input, Output;
```

```

double kp = 4, ki = 14.8, kd = 0.36;

int thermoDO = 2;
int thermoCS = 3;
int thermoCLK = 4;

PID myPID(&Input, &Output, &Setpoint, kp, ki, kd, DIRECT);
Adafruit_MAX31855 thermocouple(thermoCLK, thermoCS, thermoDO);
Adafruit_RGBLCDShield lcd = Adafruit_RGBLCDShield();

void setup()
{
    Serial.begin(9600);

    myPID.SetOutputLimits(0, 255);
    myPID.SetSampleTime(100);

    lcd.begin(16, 2);
    lcd.print("Turning on...");
    for (int i = 1; i < 8; i++) {
        lcd.setBacklight(i);
        delay(500);
    }

    analogWrite(RelayPin, 0);
    myPID.SetMode(MANUAL);
    Status = "off";
    Setpoint = 0;
    Input = thermocouple.readCelsius();

    lcd.clear();
    lastprnt = 0;
    println1(Setpoint);
    println2(Status, Input);
}

void loop()
{
    Input = thermocouple.readCelsius();

```

```

if (Input == Input) {
    while (Serial.available() > 0) {

        char received = Serial.read();

        if (received == '\n') {
            if (msg == "on") {
                myPID.SetMode(AUTOMATIC);
                Status = "On";
            } else if (msg == "off") {
                Setpoint = 0;
                analogWrite(RelayPin,0);
                myPID.SetMode(MANUAL);
                Status = "Off";
                println1(Setpoint);
            } else if (msg == "parameters?") {
                readParameters();
            } else if (msg == "temperature?") {
                Serial.println(Input);
            } else if (msg == "info") {
                String info = float2str(Input,10);
                info += "/";
                info += float2str(Output,10);
                Serial.println(info);
            } else if (msg.startsWith("set")) {
                Setpoint = setTemp(msg);
                println1(Setpoint);
            } else if (msg.startsWith("k")) {
                parseTunings(kp,ki,kd,msg);
                myPID.SetTunings(kp,ki,kd);
            } else {
                Serial.print("Error:");
                Serial.print("\t");
                Serial.println("Not a command!");
            }
            msg = "";
        } else {
            msg += received;
        }
    }
}

```

```

println2(Status,Input);

if (Status == "On") {
    if ((int)Input == (int)Setpoint) {
        lcd.setBacklight(GREEN);
    } else {
        lcd.setBacklight(WHITE);
    }
} else {
    lcd.setBacklight(RED);
}

myPID.Compute();
analogWrite(RelayPin,Output);
}

}

String float2str( double val, unsigned int precision){

    char buf[10];
    String reply;
    sprintf(buf,"%d",int(val));
    reply += buf;
    if (precision > 1) {
        reply += ".";

        unsigned int frac;
        frac = (val - int(val)) * precision;
        sprintf(buf,"%d",frac);
        reply += buf;
    }
    return reply;
}

void readParameters() {
    String parameters;
    double kp, kd, ki;

```



```

String k;

kp = myPID.GetKp();
kd = myPID.GetKd();
ki = myPID.GetKi();

k = float2str(kp,100);
parameters += k;
parameters += "/";

k= float2str(ki,100);
parameters += k;
parameters += "/";

k = float2str(kd,100);
parameters += k;

Serial.println(parameters);

}

double setTemp(String msg) {
    double setpoint;
    String temp = msg.substring(3);
    setpoint = ((double) temp.toInt());
    return setpoint;
}

void parseTunings(double &p,double &i,double &d,String msg) {
    String value = msg.substring(2);
    double kval = ((double) value.toInt())/100;
    double a = 712 / (double)100;

    switch (msg.charAt(1)) {
        case 'i':
            i = kval;
            break;
        case 'd':
            d = kval;
            break;
    }
}

```

```

        case 'p':
            p = kval;
    }
}

void println1(double setpoint) {
    String ln1;
    String setstr;

    setstr = padString(float2str(setpoint,1),5,0) + (char)223 +
        "C";

    ln1 = padString("SET:",9,1);
    ln1 += setstr;

    lcd.setCursor(0,0);
    lcd.print(ln1);
}

void println2(String Status, double Input) {
    String ln2;
    String temp;

    temp = padString(float2str(Input,1),5,0) + (char)223 + "C";
    ln2 = padString(Status,9,1);
    ln2 += temp;

    lcd.setCursor(0,1);
    lcd.print(ln2);
}

String padString(String str, int lngth, int type) {
    String padded;
    char padding;

    int diff = lngth - str.length();

    if (diff > 0) {
        switch (type) {
            case 0:
                for (int i = 1; i <= diff; i++) {

```

```

        padded += " ";
    }
    padded += str;
    break;
case 1:
    padded += str;
    for (int i = 1; i <= diff; i++) {
        padded += " ";
    }
    break;
}
} else if (diff == 0) {
    padded = str;
} else {
    padded = "ERR";
}

return padded;
}

```

“heatControl.m”

MATLAB function to set and read the temperature of the heater, as well as set the PID parameters and turn it off and on.

```

function reply = heatControl(device,cmd,value)

if (nargin < 3)
    value = 0;
end

switch cmd
    case 'temperature'
        fprintf(device,'temperature?');
        pause(0.1);
        reply = fscanf(device);
    case 'info'
        fprintf(device,'info');
        pause(0.1);
        reply = fscanf(device);
    case 'set'
        fprintf(device,['set' num2str(value)]);

```

```

        reply = value;
    case 'on'
        fprintf(device,'on');
        reply = 1;
    case 'off'
        fprintf(device,'off');
        reply = 0;
    otherwise
        reply = 0;
        fprintf(2,'NOT A COMMAND!');
end

```

“model126.m”

MATLAB function to control the Model 12-6 HPLC pump used in the HPLC system.

```

function reply = model126(serialObject,cmd,value)

    bytes = 0;

    if (nargin == 2)
        value = ' ';
    end

    if (isstr(value) == 0)
        cmd = '#';
        error('model126:noValidInput','Value must be written as
            string');
    end

    if (serialObject.BaudRate ~= 9600)
        set(serialObject,'BaudRate',9600);
    end

    if (strcmp(serialObject.FlowControl,'software') == 0)
        set(serialObject,'FlowControl','software');
    end

    switch cmd
        case 'start'
            fullcmd = 'RU';
        case 'stop'

```

```

        fullcmd = 'ST';
case 'setRate'
    fullcmd = ['FM' value];
case 'readPressure'
    fullcmd = 'PR';
case 'readPressureRate'
    fullcmd = 'CC';
case 'readSetup'
    fullcmd = 'CS';
case 'readID'
    fullcmd = 'ID';
case 'setUpperLimit'
    if (length(value) ~= 4)
        fullcmd = '#';
        fprintf(2,'ERROR:\tValue must be four digits
            long. e.g. 0900.\n');
    else
        fullcmd = ['UP' value];
    end
case 'setLowerLimit'
    if (length(value) ~= 4)
        fullcmd = '#';
        fprintf(2,'ERROR:\tValue must be four digits
            long. e.g. 0900.\n');
    else
        fullcmd = ['LP' value];
    end
case 'forceFault'
    fullcmd = 'SF';
case 'readFault'
    fullcmd = 'RF';
case 'disablePad'
    fullcmd = 'KD';
case 'enablePad'
    fullcmd = 'KE';
case 'setComp'
    if (length(value) ~= 2)
        fullcmd = '#';
        fprintf(2,'ERROR:\tValue must be two digits
            long. e.g. 09.\n');
    else

```

```

        fullcmd = ['PC' value];
    end
case 'readComp'
    fullcmd = 'RC';
case 'setHead'
    if ((length(value) ~= 1) || (strcmp(value, ' ') == 1))
        fullcmd = '#';
        fprintf(2, 'ERROR:\tValue must be one digit.\n');
    else
        fullcmd = ['HT' value];
    end
case 'readHead'
    fullcmd = 'RH';
case 'readFullSetup'
    fullcmd = 'PI';
case 'reset'
    fullcmd = 'RE';
otherwise
    fullcmd = '#';
end

fprintf(serialObject, '%s', fullcmd);

while (bytes == 0)
    bytes = serialObject.BytesAvailable;
    pause(0.01);
end
pause(0.1);

reply =
    fscanf(serialObject, '%s', serialObject.BytesAvailable);

```

“valveControl.m”

MATLAB function to control the injection valve and switching valve in the HPLC system.

```

function reply = valveControl(valve, position)

    switch position
        case 'waste'
            fprintf(valve, 'W');
        case 'recycle'

```

```

        fprintf(valve,'R');
    case 'load'
        fprintf(valve,'L');
    case 'inject'
        fprintf(valve,'I');
    otherwise
        reply = 'error';
        if (isstr(position) == 0)
            fprintf(2,'ERROR:\tPosition must be a string!');
        else
            fprintf(2,'ERROR:\tNot a recognised position');
        end
    end

end

pause(2);
reply = fscanff(valve);
end

```

“lockParams.m”

MATLAB function to return the relevant parameters for the software-based lock-in amplification of signal from the photodetectors in the analyser of the HPLC system.

```

function [fsChan nsChan fMod A] =
    lockParams(sampleRate,totalSamples,nChannels,kFactor)
    fsChan = sampleRate/nChannels;
    nsChan = totalSamples/nChannels;
    fMod = (kFactor*fsChan)/nsChan;
    x = exp((fMod/fsChan)*2*pi*(0 + 1i));
    y = -1.*(1:nsChan);
    A = x.^y;

```

“sqwave.m”

MATLAB function to generate a square wave for the software-based lock-in amplification.

```

function output = sqwave(nSamples, sampleRate, frequency,
    amplitude, offset, initPhase,duty)
    output = zeros(nSamples,1);
    phase = initPhase + ([0:(nSamples - 1)] ./ sampleRate) .*
        frequency .* 360;
    pmod = mod(phase,360);

```

```

for i = 1:nSamples
    if(pmod(i) >= 0 & pmod(i) <(0.01*duty*360))
        output(i) = 1;
    elseif(pmod(i) >= (0.01*duty*360) & pmod(i) < 360)
        output(i) = -1;
    end
end

output = (output.*amplitude) + offset;

```

“bareProcess.m”

MATLAB function called by an event listener on the HPLC system DAQ card, for applying the software-based lock-in amplification to each second of signal acquired by the analyser.

```

function bareProcess(timestamp,data,fid,A,elutionTime,detector)

% bare-bones data processing for hplc. Performs software
% lock-in and saves to file

time = timestamp(1);

if (time > elutionTime)
    detector.stop();
    fclose(fid);
else
    ampData = abs(A*data);
    normData = ampData(1)/ampData(2);
    status = detector.IsDone;
    fprintf(fid,[num2str(time,16) '\t' num2str(normData,16)
        '\n']);
end

end

```

“chromatogram.m”

MATLAB function for processing the signal from the in-line HPLC system into a chromatogram.

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```



```

%%% Function to process raw signal into a chromatogram %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function chrom = chromatogram(time,signal)

[m,n] = size(signal);

if (size(time,2) < n)
    time = repmat(time,1,n);
end

%%%%%%%%% Calculate reference signal and absorbance %%%%%%%%%

io = mean(signal(50:150,:));
absorbance = -log10(signal./repmat(io,m,1));

%%%%%%%%% Correct sloping baselines %%%%%%%%%

[~,sol_peak] = max(absorbance(1:220,:));

[~,middle_peak] = max(absorbance(410:540,:));
middle_peak = middle_peak + 409;

baseline = zeros(m,n);

window = 5;
for j = 1:n

    if middle_peak+100 < m-15*window-3
        base_points = round([linspace(50,150,5),
            linspace(middle_peak(j)-35,middle_peak(j)-20,5),
            linspace(middle_peak(j)+60,middle_peak(j)+100,5),
            linspace(m-10*window-3,m-window-1,5)]);
    else
        base_points = round([linspace(50,150,5),
            linspace(middle_peak(j)-35,middle_peak(j)-20,5),
            linspace(middle_peak(j)+60,m-10*window-4,5),
            linspace(m-10*window-3,m-window-1,5)]);
    end

    n_point = length(base_points);

```

```

base_abs = zeros(n_point,1);
base_time = zeros(n_point,1);

for i = 1:n_point
    base_abs(i) = mean(absorbance(base_points(i) -
        window:base_points(i) + window, j));
    base_time(i) = mean(time(base_points(i) -
        window:base_points(i) + window, j));
end

baseline(:,j) = interp1(base_time,base_abs,time(:,j));

end

% Subtract baseline from absorbance
chrom = absorbance - baseline;

% Normalise signal to solvent peak
sol_peak = repmat(sol_peak,51,1) + repmat([-20:30]',1,n) +
    repmat([0:n-1],51,1).*m; %'
sol_area = trapz(abs(chrom(sol_peak)));

chrom = chrom./repmat(sol_area,m,1);

plot(time, chrom)

```

“findheights.m”

MATLAB function for finding the adduct peaks in a chromatogram, and returning their retention time and height.

```

function [peak_time,peak_height] = findheights(time,signal)

[m,n] = size(signal);
if m > 1201
    start = [1200; 800; 580; 400];
    finish = [m; 1000; 800; 580];
elseif m > 901
    start = [900; 600; 450; 350];
    finish = [m; 700; 550; 450];
else

```

```

    start = [600; 410; 290; 230];
    finish = [m; 540; 400; 280];
end

peak_time = zeros(4,n);

for i = 1:4
    peak_range = start(i):finish(i);
    [peak_height(:,i),peak_pos] = max(signal(peak_range, :, :));
    peak_time(i,:) = time(peak_pos+start(i));
end

peak_height = peak_height';
peak_time = peak_time';

```

“convertpeaks.m”

MATLAB function for converting a list of adduct peak heights into concentrations of adducts, and then mole fractions, using calibration constants stored in a file.

```

function [concentration,mf] = convertpeaks(y)

[m,n] = size(y);

y(isnan(y)) = 0;

mr = [720; 824; 928; 1032];

fid = fopen('weighted_calibration_file.csv');
HDRS = textscan(fid,'%s %s %s %s %s %s %s %s %s %s %s %s',1,'delimiter',' ','');
DATA = textscan(fid,'%s %f %f %f %f %f %f %f %f %f %f %f',1,'delimiter',' ','');
fclose(fid);

a = DATA{2};
b = DATA{3};

x = (y - repmat(a,1,n))./repmat(b,1,n);

x(x < 0) = 0;
x(isnan(x)) = 0;

```

```

concentration = x./repmat(mr,1,n); % concentration in mmol/mL
mf = concentration./repmat(sum(concentration),4,1); % mole
    fraction
mf(isnan(mf)) = 0;

```

Code for Control of the Automated Flow Reactor

“runReactor.m”

MATLAB function that can be used to carry out a synthesis on the automated flow reactor. NB the following code uses the term ‘reaction time’ in place of the more correct ‘residence time’.

```

function mf = runReactor(fr1,fr2,T,filestem)

    % Set up reactor parameters

    sultine_rate = fr2; % Sultine flow rate in uL/min
    c60_rate = fr1; % C60 flow rate in uL/min
    total_rate = fr1 + fr2;
    n = length(total_rate);
    stock_c = [2,1.4]/1000; % stock concentrations in mg/uL
    mr = [720,168]; % molar mass of reagents

    reactor_volume = 935; % Reactor volume in uL
    RT = reactor_volume./total_rate; % Reaction time in minutes

    % Set up HPLC parameters

    sample_volume = 5; % HPLC sample-coil volume in uL
    elution_time = 13*60; % HPLC elution time in seconds

    [fsChan, nsChan, fMod, A] = lockParams(250000,250000,2,113);
    % parameters for lock-in

    output = [sqwave(nsChan, fsChan, fMod, 2.25, 2.25, 0, 50),
        linspace(10, 10, nsChan)']; % 'LED square wave pulse
    turnOn = [linspace(4.5,4.5,1000)',linspace(0,10,1000)']; %
        turn on photodetectors
    turnOff = [linspace(4.5,4.5,1000)',linspace(10,0,1000)']; %

```

```

    turn off photodetectors

% Generate file names
hplc_names = cell(n,1);
temp_names = cell(n,1);

reactor_folder = ['HPLC\',datestr(date,'yymmdd')];
[filepath, n_start] = fileNameNum(filestem,reactor_folder);

padding = '000';

for i = 1:n
    n_pad = length(padding) - length(num2str(i+n_start));
    pad = padding(1:n_pad);
    file_num = [pad,num2str(i+n_start)];
    hplc_names(i) = {strcat(filepath, filestem, '.hplc.',
        num2str(file_num), '.txt')};
    temp_names(i) = {strcat(filepath, filestem, '.temp.',
        num2str(file_num), '.txt')};
end

log_name = {strcat(filepath,filestem,'.log.txt')};
output_name = {strcat(filepath, filestem, '.output.',
    num2str(1+n_start), '.txt')};

% Start hardware
heater = serial('COM14');
valve = serial('COM17');
hplc_pump = serial('COM16');

fopen(heater); % open serial objects
fopen(valve);
    fopen(hplc_pump);
pause(7);

    fprintf(heater,'kp3000'); fprintf(heater,'ki20');
    fprintf(heater,'kd0600'); % ensure pid parameters are right
heatControl(heater,'on'); % turn heater on, setpoint at 0
valvePosition = valveControl(valve,'inject'); % put switching
    valve to correct position

```

```

% Start running points

% Open log file
log_id = fopen(char(log_name), 'at');

% Start timer clock
now = tic;

% Set heater to 1st temperature
fprintf('heater set to %f\n', T(1))
heatControl(heater, 'set', T(1));
heat_start = toc(now);

% Wait for heater to reach temperature
temperature = str2num(heatControl(heater, 'temperature'));
temp_hist = [];
isconstant = 0;
while ~isconstant
    temperature = str2num(heatControl(heater, 'temperature'));
    fprintf(log_id, '%f,%f\n', datenum(datetime('now')),
        temperature);

    if length(temp_hist) < 200
        temp_hist = [temp_hist; temperature];
    else
        temp_hist = [temp_hist(2:end); temperature];
        temp_std = std(temp_hist);
        temp_r = abs(T(1) - temperature);

        if (temp_r < 1 & temp_std < 1)
            isconstant = 1;
        end
    end

    pause(1.5);
end

for i = 1:n
    % For each data point:

    hplc_id = fopen(char(hplc_names(i)), 'wt'); % open hplc file

```

```

        for writing

if i == 1
    fprintf('c60 rate is %f\n', c60_rate(i))
    fprintf('sultine rate is %f\r\n', sultine_rate(i))
    syrris('ch1FR',c60_rate(i),'ch2FR',sultine_rate(i));
    syrris('ch1infuse',true,'ch2infuse',true)
    datapoint_start = toc(now);
    temp_id = fopen(char(temp_names(i)),'wt');
    % if it is the first point, set flow rates, start pumps and
    % point clock
end

while (toc(now) - datapoint_start < RT(i)*60*2)
    % Wait for twice the reaction time
    % Poll temperature and save to log file and temperature
    % file
    temperature = str2num(heatControl(heater,'temperature'));
    fprintf(log_id, '%f,%f\n', datenum(datetime('now')),
        temperature);
    fprintf(temp_id, '%f,%f\n', (toc(now)-datapoint_start),
        temperature);
    pause(1.5);
end

fclose(temp_id);

% Set up HPLC
detector = daq.createSession('ni');
detector.addAnalogOutputChannel('Dev2',0,'Voltage'); % LED
% power
detector.addAnalogOutputChannel('Dev2',1,'Voltage'); %
% photodiode powers
detector.addAnalogInputChannel('Dev2',1,'Voltage'); % end
% photodiode
detector.addAnalogInputChannel('Dev2',2,'Voltage'); % side
% photodiode

detector.queueOutputData(turnOn); % turn on photodiodes
detector.startForeground();

```

```

detector.addTriggerConnection('External', 'Dev2/PFI3',
    'StartTrigger');
detector.Connections(1).TriggerCondition = 'FallingEdge'; %
    switch valve triggering

detector.Rate = fsChan;
detector.IsContinuous = true;

refresh = detector.addlistener('DataRequired',
    @(src,event)detector.queueOutputData(output));
saver = detector.addlistener('DataAvailable',
    @(src,event)bareProcess(event.TimeStamps, event.Data,
        hplc_id, A, elution_time, detector));

detector.NotifyWhenDataAvailableExceeds = nsChan;
detector.NotifyWhenScansQueuedBelow = nsChan*2;
detector.queueOutputData([output;output]);

% Load reaction mixture into sample coil
fprintf('\r\nLoading Sample...\r\n')
load_time = 600*sample_volume/total_rate(i);
valvePosition = valveControl(valve,'load');
pause(load_time);
syrris('ch1infuse',false,'ch2infuse',false); % stop pumps to
    conserve fluids

valvePosition = valveControl(valve,'waste');

% Start HPLC
detector.startBackground();
fprintf('HPLC STARTED!\r\n')
valvePosition = valveControl(valve,'inject');

    % wait, then check if hplc was triggered properly
    pause(6);
    if ~detector.IsRunning
        heatControl(heater,'set',23);
        heatControl(heater,'off');
        model126(hplc_pump,'stop');
        %send_to_gmail('Rheodyne valve trigger
            missed','ERROR: Valve');

```



```

        error('ERROR: Rheodyne valve error')
    end

    % if there are points left, start the next point while HPLC
    runs
    if (i < n)
        % start heating clock
        heat_start = toc(now);
        fprintf('heater set to %f\n', T(i+1))
        heatControl(heater, 'set', T(i+1));

        % Wait for heater to reach temperature
        if T(i + 1) == T(i)
            fprintf('Same temperature!\n')
        else
            temp_hist = [];
            isconstant = 0;
            while ~isconstant
                temperature =
                    str2num(heatControl(heater, 'temperature'));
                fprintf(log_id, '%f,%f\n',
                    datenum(datetime('now')), temperature);

                if length(temp_hist) < 200
                    temp_hist = [temp_hist; temperature];
                else
                    temp_hist = [temp_hist(2:end); temperature];
                    temp_std = std(temp_hist);
                    temp_r = abs(T(i + 1) - temperature);

                    if (temp_r < 1 & temp_std < 1)
                        isconstant = 1;
                    end
                end

            pause(1.5);
        end
    end

    % Wait for reaction time to be equal to time left on HPLC
    heat_time = toc(now) - heat_start;

```

```

while (elution_time - heat_time > RT(i+1)*60*2)
    % Poll temperature and save to log and temperature files
    heat_time = toc(now) - heat_start;
    temperature = str2num(heatControl(heater,
        'temperature'));
    fprintf(log_id, '%f,%f\n', datenum(datetime('now')),
        temperature);
    pause(1.5);
end

% Start pumping
fprintf('PUMPS STARTED\n');
syrris('ch1FR', c60_rate(i+1), 'ch2FR', sultine_rate(i+1),
    'ch1infuse', true, 'ch2infuse', true);
temp_id = fopen(char(temp_names(i+1)), 'wt');
datapoint_start = toc(now);

% Wait for HPLC to finish
fprintf('Waiting for HPLC to finish...\n')
while (heat_time < elution_time)
    % log temperature
    heat_time = toc(now) - heat_start;
    temperature = str2num(heatControl(heater, 'temperature'));
    fprintf(log_id, '%f,%f\n', datenum(datetime('now')),
        temperature);
    fprintf(temp_id, '%f,%f\n', (toc(now) -
        datapoint_start), temperature);
    pause(1.5);
end

% Another while loop to ensure reactor does not move on
% before hplc is done
while (detector.IsDone == 0)
    % log temperature
    heat_time = toc(now) - heat_start;
    temperature = str2num(heatControl(heater, 'temperature'));
    fprintf(log_id, '%f,%f\n', datenum(datetime('now')),
        temperature);
    fprintf(temp_id, '%f,%f\n', (toc(now)-datapoint_start),
        temperature);
    pause(1.5);
end

```

```

        end

    else
        % if it is the last iteration, what until hplc is done
        while (detector.IsDone == 0)
            pause(1.5);
        end
    end

    % Turn off HPLC
    fprintf('HPLC OFF!\n')
    valvePosition = valveControl(valve,'recycle');

    detector.removeConnection(1);
    detector.removeChannel(3:4);
    detector.IsContinuous = false;

    delete(refresh);
    delete(saver);

    detector.queueOutputData(turnOff);
    detector.startForeground();

    clear refresh saver detector
end

% Turn everything off
fclose(log_id);
heatControl(heater,'set',23);
heatControl(heater,'off');

fclose(heater);
fclose(valve);
fclose(hplc_pump);

% Load hplc files

time = zeros(elution_time,n);
signal = zeros(elution_time,n);

for i = 1:n

```

```

data = load(char(hplc_names(i)));
if length(data) < elution_time
    t = [data(end,1)+1:elution_time-1]';
    s = repmat(mean(data(end-10:end,2)),(elution_time -
        length(data)),1);
    data = [data;t,s];
end
time(:,i) = data(1:elution_time,1);
signal(:,i) = data(1:elution_time,2);
end

% Convert to chromatogram
c = chromatogram(time,signal);

% Find and integrate peaks
[peak_time,peak_height] = findheights(time,c);

% Calculate mole fraction
[concentration,mf] = convertpeaks(peak_height);
mf = mf';

molar_rate =
    [c60_rate,sultine_rate].*repmat(stock_c,n,1)./repmat(mr,n,1);
XS = molar_rate(:,2)./molar_rate(:,1); % molar ratio of
    sultine:C60
output = [RT, XS, T, mf];

save(char(output_name),'output','-ascii');

end

```

“grid_experiment.m”

Top level MATLAB script that calls the “runReactor.m” function, used to run the parameter space mapping experiment in Chapter 1.

```

close all;
clear
clc

% Choose points to look at
T = linspace(100,150,6)';

```

```

RT = linspace(3,31,6)';
XS = linspace(1.5,6,6)';
% Generate grid of points
[x,y,z] = meshgrid(RT,XS,T);
% Convert RT and XS into flow rates
fr_total = 922./x(:);
ratio = y(:)./3;
fr_c = fr_total./(1 + ratio);
fr_s = fr_total - fr_c;
T = z(:);
% randomise the order
%runorder = randperm(length(T))';
runorder = load('grid_6_runorder.txt');
% Make empty matrix for mole fraction data
mf = zeros(246,4);
filestem = 'bw160516';

for i = 1:6
    % Calculate start and stop indicies
    start = (i - 1)*36 + 1;
    stop = i * 36;
    % Generate repeats from points already looked at
    repeats = runorder(randi([1 stop],5,1));
    points_to_run = [runorder(start:stop);repeats];
    % call Reactor and feed data
    mf(points_to_run,:) = runReactor(fr_c(points_to_run),
        fr_s(points_to_run), T(points_to_run), filestem);
end

```

Code used for Automated Optimisations

The package developed to provide a class-based interface to the SNOBfit optimisation algorithm is available at: https://github.com/jdmgroup/SNOBfit_for_chemical_optimisations. Below are examples of objective and constraint function definitions, and scripts to run unconstrained and constrained optimisations.

“3d_objective_function.m”

An example of an objective function definition used to run a SNOBfit optimisation using the class-based interface and the automated reactor.

```

function f = 3d_objective_function(SNOB)
    % Example of an objective function tagetting maximum X2
    % by varying 3 reaction parameters

    x = SNOB.next;

    fr1 = x(:,1);
    fr2 = x(:,2);
    T = x(:,3);

    mf = runReactor(fr1,fr2,T,SNOB.name);

    X2 = mf(:,3);
    f = 1 - X2;

end

```

“3d_constraint_function.m”

An example of a constraint function definition used to run a SNOBfit optimisation using the class-based interface and the automated reactor.

```

function F = 3d_constraint_function(SNOB)
    % Example of a constraint function to be used with the
    % previous example
    % of an objective function, with constraints on the amount
    % of product
    % composed of X1 and X2, and the ratio of X1 to X2

    filestem = SNOB.name;
    hplc_root = 'C:/Users/jdmgroup/Dropbox/PhD/HPLC/';

    % find the last hplc folder
    files = dir(hplc_root);
    isfolder = [files.isdir];
    folders = files(isfolder);
    [~,order] = sort([folders(:).datenum]);
    folder_names = {folders(order).name};
    last_folder = folder_names{end};

    % find all the output files

```

```

outputs =
    dir([hplc_root,last_folder,'/',filestem,'.output*.txt']);
[~,order] = sort([outputs(:).datenum]);
output_names = {outputs(order).name};

% load in the last output file
data = load([hplc_root,last_folder,'/',output_names{end}]);
mf = data(:,4:end);

X2 = mf(:,3)
X1 = mf(:,1)

% function value is mole fraction of oxcta
F(:,1) = 1 - (X2 + X1);
F(:,2) = X1./X2;

end

```

“unconstrained_optimisation.m”

An example of a script used to run an unconstrained SNOBfit optimisation using the class-based interface and the automated reactor.

```

% Example of a script to run a constrained optimisation
snobfit_object = snobfitclass();

% set the name and name of the objective function
snobfit_object.name = 'unconstrained_optimisation';
snobfit_object.fcn = '3d_objective_function';

% set the bounds, with linked bounds for the two reagent flow
    rates
snobfit_object.linked = true;
snobfit_object.xyMin = 90;
snobfit_object.xyMax = 922;
snobfit_object.minRatio = 0.5;
snobfit_object.maxRatio = 2;
snobfit_object.zMin = 100;
snobfit_object.zMax = 150;

% set the termination condition to total number of function
    evaluations

```

```
snobfit_object.termination = 'n_runs';
snobfit_object.ncall = 100;

% start the optimisation
snobfit_object.startExp;
```

“constrained_optimisation.m”

An example of a script used to run an constrained SNOBFit optimisation using the class-based interface and the automated reactor.

```
% Example of a script to run a constrained optimisation
snobfit_object = snobfitclass();

% set the name and name of the objective function
snobfit_object.name = 'constrained_optimisation';
snobfit_object.fcn = '3d_objective_function';

% set the name of the constraint function
snobfit_object.constrained = true;
snobfit_object.constraintFcn = '3d_constraint_function';

% set the limits of constraints
snobfit_object.F_lower = [0.9; 0];
snobfit_object.F_upper = [1.0; 0.5];

% set the bounds, with linked bounds for the two reagent flow
  rates
snobfit_object.linked = true;
snobfit_object.xyMin = 90;
snobfit_object.xyMax = 922;
snobfit_object.minRatio = 0.5;
snobfit_object.maxRatio = 2;
snobfit_object.zMin = 100;
snobfit_object.zMax = 150;

% set the termination condition to total number of function
  evaluations
snobfit_object.termination = 'n_runs';
snobfit_object.ncall = 100;

% start the optimisation
```



```
snobfit_object.startExp;
```

Rights and Permissions

Below are details of the permissions granted for the reproduction of some figures and images in this thesis.

Figure 1.2

This was reproduced from Duraiswamy and Khan²⁸ with permission from Wiley-VCH. Proof of permission is below.



Thank you for your order!

Dear Barnaby Walker,

Thank you for placing your order through Copyright Clearance Center's RightsLink[®] service.

Order Summary

Licensee:	Barnaby Walker
Order Date:	Sep 24, 2017
Order Number:	4195631445395
Publication:	Small
Title:	Droplet-Based Microfluidic Synthesis of Anisotropic Metal Nanocrystals
Type of Use:	Dissertation/Thesis
Order Total:	0.00 GBP

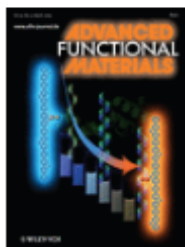
View or print complete [details](#) of your order and the publisher's terms and conditions.

Sincerely,

Copyright Clearance Center

Figure 1.4

This was reproduced from Bannock *et al*¹⁹ with permission from Wiley-VCH. Proof of permission is below.



Thank you for your order!

Dear Barnaby Walker,

Thank you for placing your order through Copyright Clearance Center's RightsLink[®] service.

Order Summary

Licensee: Barnaby Walker
Order Date: Sep 24, 2017
Order Number: 4195640168728
Publication: Advanced Functional Materials
Title: Continuous Synthesis of Device-Grade Semiconducting Polymers in Droplet-Based Microreactors
Type of Use: Dissertation/Thesis
Order Total: 0.00 GBP

View or print complete [details](#) of your order and the publisher's terms and conditions.

Sincerely,

Copyright Clearance Center

Figure 5.3

This was reproduced from Nightingale *et al*⁵⁷, and is licensed by the Nature Publishing Group under a Creative Commons CC-BY 3.0 license.

Figure 5.2

This was reproduced from Zhen Li *et al*¹⁰², using the attribution detailed in the permission granted by the American Chemical Society. Proof of this is shown below.



[Home](#) [Create Account](#) [Help](#) 

 **ACS Publications**
Most Trusted. Most Cited. Most Read.

Title: One-Pot Reaction to Synthesize Water-Soluble Magnetite Nanocrystals

Author: Zhen Li, Hui Chen, Haobo Bao, et al

Publication: Chemistry of Materials

Publisher: American Chemical Society

Date: Apr 1, 2004

Copyright © 2004, American Chemical Society

LOGIN

If you're a [copyright.com](#) user, you can login to RightsLink using your copyright.com credentials. Already a [RightsLink](#) user or want to [learn more?](#)

PERMISSION/LICENSE IS GRANTED FOR YOUR ORDER AT NO CHARGE

This type of permission/license, instead of the standard Terms & Conditions, is sent to you because no fee is being charged for your order. Please note the following:

- Permission is granted for your request in both print and electronic formats, and translations.
- If figures and/or tables were requested, they may be adapted or used in part.
- Please print this page for your records and send a copy of it to your publisher/graduate school.
- Appropriate credit for the requested material should be given as follows: "Reprinted (adapted) with permission from (COMPLETE REFERENCE CITATION). Copyright (YEAR) American Chemical Society." Insert appropriate information in place of the capitalized words.
- One-time permission is granted only for the use specified in your request. No additional uses are granted (such as derivative works or other editions). For any other uses, please submit a new request.

If credit is given to another source for the material you requested, permission must be obtained from that source.

Bibliography

- [1] A. Kirschning, W. Solodenko and K. Mennecke, *Chem. Eur. J.*, 2006, **12**, 5972–5990.
- [2] K. F. Jensen, *AIChE J.*, 2017, **63**, 858–869.
- [3] J. Britton and C. L. Raston, *Chem. Soc. Rev.*, 2017, **46**, 1250–1271.
- [4] A. Adamo, R. L. Beingessner, M. Behnam, J. Chen, T. F. Jamison, K. F. Jensen, J.-C. M. Monbaliu, A. S. Myerson, E. M. Revalor, D. R. Snead, T. Stelzer, N. Weeranoppanant, S. Y. Wong and P. Zhang, *Science*, 2016, **352**, 61–67.
- [5] K. Jähnisch, V. Hessel, H. Löwe and M. Baerns, *Angew. Chem. Int. Ed.*, 2004, **43**, 406–446.
- [6] A. M. Nightingale and J. C. deMello, *J. Mater. Chem.*, 2010, **20**, 8454–8463.
- [7] J. H. Bannock, S. H. Krishnadasan, M. Heeney and J. C. deMello, *Mater. Horizons*, 2014, **1**, 373–378.
- [8] J.-S. Poh, D. L. Browne and S. V. Ley, *React. Chem. Eng.*, 2016, **1**, 101–105.
- [9] S. H. Lau, A. Galván, R. R. Merchant, C. Battilocchio, J. A. Souto, M. B. Berry and S. V. Ley, *Org. Lett.*, 2015, **17**, 3218–3221.
- [10] J. A. Souto, R. A. Stockman and S. V. Ley, *Org. Biomol. Chem.*, 2015, **13**, 3871–3877.
- [11] Z. Qian, I. R. Baxendale and S. V. Ley, *Chem. Eur. J.*, 2010, **16**, 12342–12348.
- [12] S. Newton, C. F. Carter, C. M. Pearson, L. De C. Alves, H. Lange, P. Thansandote and S. V. Ley, *Angew. Chem. Int. Ed.*, 2014, **53**, 4915–4920.
- [13] P. R. Murray, D. L. Browne, J. C. Pastre, C. Butters, D. Guthrie and S. V. Ley, *Org. Process Res. Dev.*, 2013, **17**, 1192–1208.
- [14] D. Lücke, T. Dalton, S. V. Ley and Z. E. Wilson, *Chem. Eur. J.*, 2016, **22**, 4206–4217.
- [15] M. Baumann, I. R. Baxendale, S. V. Ley, N. Nikbin, C. D. Smith and J. P. Tierney, *Org. Biomol. Chem.*, 2008, **6**, 1577–1586.

- [16] H. Seyler, J. Subbiah, D. J. Jones, A. B. Holmes and W. W. H. Wong, *Beilstein J. Org. Chem.*, 2013, **9**, 1492–1500.
- [17] H. Seyler, W. W. H. Wong, D. J. Jones and A. B. Holmes, *J. Org. Chem.*, 2011, **76**, 3551–3556.
- [18] H. Seyler, D. J. Jones, A. B. Holmes and W. W. H. Wong, *Chem. Commun.*, 2012, **48**, 1598–1600.
- [19] J. H. Bannock, S. H. Krishnadasan, A. M. Nightingale, C. P. Yau, K. Khaw, D. Burkitt, J. J. M. Halls, M. Heeney and J. C. deMello, *Adv. Funct. Mater.*, 2013, **23**, 2123–2129.
- [20] F. Bally, C. A. Serra, V. Hessel and G. Hadziioannou, *Chem. Eng. Sci.*, 2011, **66**, 1449–1462.
- [21] S. Krishnadasan, R. J. C. Brown, A. J. deMello and J. C. deMello, *Lab Chip*, 2007, **7**, 1434–1441.
- [22] A. M. Nightingale, S. H. Krishnadasan, D. Berhanu, X. Niu, C. Drury, R. McIntyre, E. Valsami-Jones and J. C. deMello, *Lab Chip*, 2011, **11**, 1221–1227.
- [23] J. Wang, H. Zhao, Y. Zhu and Y. Song, *J. Phys. Chem. C*, 2017, **121**, 3567–3572.
- [24] A. M. Nightingale, J. H. Bannock, S. H. Krishnadasan, F. T. F. O’Mahony, S. A. Haque, J. Sloan, C. Drury, R. McIntyre and J. C. deMello, *J. Mater. Chem. A*, 2013, **1**, 4067–4076.
- [25] T. Akdas, M. Haderlein, J. Walter, B. Apeleo Zubiri, E. Spiecker and W. Peukert, *RSC Adv.*, 2017, **7**, 10057–10063.
- [26] C. Yang, G. Feng, S. Wang, J. Dai, Y. Zhang and S. Zhou, *Chem. Phys. Lett.*, 2017, **684**, 409–413.
- [27] Y. Song, H. Modrow, L. L. Henry, C. K. Saw, E. E. Doomes, V. Palshin, J. Hormes and C. S. Kumar, *Chem. Mater.*, 2006, **18**, 2817–2827.
- [28] S. Duraiswamy and S. A. Khan, *Small*, 2009, **5**, 2828–2834.
- [29] V. Sebastian Cabeza, S. Kuhn, A. A. Kulkarni and K. F. Jensen, *Langmuir*, 2012, **28**, 7007–7013.
- [30] J. Boleininger, A. Kurz, V. Reuss and C. Sönnichsen, *Phys. Chem. Chem. Phys.*, 2006, **8**, 3824–3827.
- [31] X. Z. Lin, A. D. Terepka and H. Yang, *Nano Lett.*, 2004, **4**, 2227–2232.
- [32] L. L. Lazarus, C. T. Riche, B. C. Marin, M. Gupta, N. Malmstadt and R. L. Brutchey, *ACS Appl. Mater. Interfaces*, 2012, **4**, 3077–3083.
- [33] J. M. Köhler, M. Held, U. Hübner and J. Wagner, *Chem. Eng. Technol.*, 2007, **30**, 347–354.
- [34] O. Okafor, A. Weilhard, J. A. Fernandes, E. Karjalainen, R. Goodridge and V. Sans, *React. Chem. Eng.*, 2017, **2**, 129–136.

- [35] Y. H. Kim, L. Zhang, T. Yu, M. Jin, D. Qin and Y. Xia, *Small*, 2013, **9**, 3462–3467.
- [36] J. B. Wacker, I. Lignos, V. K. Parashar and M. A. M. Gijs, *Lab Chip*, 2012, **12**, 3111–3116.
- [37] L. Zhang, Y. Li, Q. Zhang, G. Shi and H. Wang, *Chem. Lett.*, 2011, **40**, 1371–1373.
- [38] T. H. Eun, S. H. Kim, W. J. Jeong, S. J. Jeon, S. H. Kim and S. M. Yang, *Chem. Mater.*, 2009, **21**, 201–203.
- [39] S. Li, P. M. Günther and J. M. Köhler, *J. Chem. Eng. Jpn*, 2009, **42**, 338–345.
- [40] K. Shiba and M. Ogawa, *Chem. Commun.*, 2009, **0**, 6851–6853.
- [41] K. Kumar, A. M. Nightingale, S. H. Krishnadasan, N. Kamaly, M. Wylenzinska-Arridge, K. Zeissler, W. R. Branford, E. Ware, A. J. deMello and J. C. deMello, *J. Mater. Chem.*, 2012, **22**, 4704–4708.
- [42] J. H. Bannock, M. Al-Hashimi, S. H. Krishnadasan, J. J. M. Halls, M. Heeney and J. C. deMello, *Mater. Horizons*, 2014, **1**, 214–218.
- [43] J. H. Bannock, W. Xu, T. Baïssas, M. Heeney and J. C. deMello, *Eur. Polym. J.*, 2016, **80**, 240–246.
- [44] P. A. Bayliss, I. A. Ibarra, E. Pérez, S. Yang, C. C. Tang, M. Poliakoff and M. Schröder, *Green Chem.*, 2014, **16**, 3796–3802.
- [45] M. Faustini, J. Kim, G. Y. Jeong, J. Y. Kim, H. R. Moon, W. S. Ahn and D. P. Kim, *J. Am. Chem. Soc.*, 2013, **135**, 14619–14626.
- [46] I. Lignos, S. Stavrakis, G. Nedelcu, L. Protesescu, A. J. deMello and M. V. Kovalenko, *Nano Lett.*, 2016, **16**, 1869–1877.
- [47] H. Yang, W. Luan, Z. Wan, S. T. Tu, W. K. Yuan and Z. M. Wang, *Cryst. Growth Des.*, 2009, **9**, 4807–4813.
- [48] T. F. Abelha, T. W. Phillips, J. H. Bannock, A. M. Nightingale, C. A. Dreiss, E. Kemal, L. Urbano, J. C. deMello, M. Green and L. A. Dailey, *Nanoscale*, 2017, **9**, 2009–2019.
- [49] W. K. Wong, S. K. Yap, Y. C. Lim, S. A. Khan, F. Pelletier and E. C. Corbos, *React. Chem. Eng.*, 2017, **2**, 636–641.
- [50] J. Wagner and J. M. Köhler, *Nano Lett.*, 2005, **5**, 685–691.
- [51] E. M. Chan, A. P. Alivisatos and R. A. Mathies, *J. Am. Chem. Soc.*, 2005, **127**, 13854–13861.
- [52] M. Schoenitz, L. Grundemann, W. Augustin and S. Scholl, *Chem. Commun.*, 2015, **51**, 8213–8228.
- [53] J. Sedelmeier, S. V. Ley, I. R. Baxendale and M. Baumann, *Org. Lett.*, 2010, **12**, 3618–3621.
- [54] S. A. Khan and S. Duraiswamy, *Lab Chip*, 2012, **12**, 1807–1812.

- [55] S. Duraiswamy and S. A. Khan, *Nano Lett.*, 2010, **10**, 3757–3763.
- [56] E. Y. Erdem, J. C. Cheng, F. M. Doyle and A. P. Pisano, *Small*, 2014, **10**, 1076–1080.
- [57] A. M. Nightingale, T. W. Phillips, J. H. Bannock and J. C. deMello, *Nat. Commun.*, 2014, **5**, 3777–3784.
- [58] S. Marre, J. Park, J. Rempel, J. Guan, M. G. Bawendi and K. F. Jensen, *Adv. Mater.*, 2008, **20**, 4830–4834.
- [59] D. C. Fabry, E. Sugiono and M. Rueping, *React. Chem. Eng.*, 2016, **1**, 129–133.
- [60] V. Sans and L. Cronin, *Chem. Soc. Rev.*, 2016, **45**, 2032–2043.
- [61] V. Sans, S. Glatzel, F. J. Douglas, D. A. Maclaren, A. Lapkin and L. Cronin, *Chem. Sci.*, 2014, **5**, 1153–1157.
- [62] H. Lu, M. A. Schmidt and K. F. Jensen, *Lab Chip*, 2001, **1**, 22–28.
- [63] S. T. Müller, A. Murat, D. Maillos, P. Lesimple, P. Hellier and T. Wirth, *Chem. Eur. J.*, 2015, **21**, 7016–7020.
- [64] J. S. Moore and K. F. Jensen, *Angew. Chem. Int. Ed.*, 2014, **53**, 470–473.
- [65] C. B. Minnich, L. Greiner, C. Reimers, M. Uerdingen and M. A. Liauw, *Chem. Eng. J.*, 2011, **168**, 759–764.
- [66] T. Brodmann, P. Koos, A. Metzger, P. Knochel and S. V. Ley, *Org. Process Res. Dev.*, 2012, **16**, 1102–1113.
- [67] C. J. Smith, N. Nikbin, S. V. Ley, H. Lange and I. R. Baxendale, *Org. Biomol. Chem.*, 2011, **9**, 1938–1947.
- [68] A. E. Cervera-Padrell, J. P. Nielsen, M. Jønch Pedersen, K. Müller Christensen, A. R. Mortensen, T. Skovby, K. Dam-Johansen, S. Kiil and K. V. Gernaey, *Org. Process Res. Dev.*, 2012, **16**, 901–914.
- [69] S.-A. Leung, R. F. Winkle, R. C. R. Wootton and A. J. deMello, *Analyst*, 2005, **130**, 46–51.
- [70] T. A. Hamlin and N. E. Leadbeater, *Beilstein J. Org. Chem.*, 2013, **9**, 1843–1852.
- [71] S. Mozharov, A. Nordon, D. Littlejohn, C. Wiles, P. Watts, P. Dallin and J. M. Girkin, *J. Am. Chem. Soc.*, 2011, **133**, 3601–3608.
- [72] V. Sans, L. Porwol, V. Dragone and L. Cronin, *Chem. Sci.*, 2015, **6**, 1258–1264.
- [73] T. H. Rehm, C. Hofmann, D. Reinhard, H.-J. Kost, P. Löb, M. Besold, K. Welzel, J. Barten, A. Didenko, D. V. Sevenard, B. Lix, A. R. Hillson and S. D. Riegel, *React. Chem. Eng.*, 2017, **2**, 315–323.
- [74] S. K. Küster, E. Danieli, B. Blümich and F. Casanova, *Phys. Chem. Chem. Phys.*, 2011, **13**, 13172–13176.

- [75] M. V. Gomez, A. M. Rodriguez, A. De La Hoz, F. Jimenez-Marquez, R. M. Fratila, P. A. Barneveld and A. H. Velders, *Anal. Chem.*, 2015, **87**, 10547–10555.
- [76] M. V. Gomez, H. H. J. Verputten, A. Díaz-Ortíz, A. Moreno, A. de la Hoz and A. H. Velders, *Chem. Commun.*, 2010, **46**, 4514–4516.
- [77] S. Krishnadasan, J. Tovilla, R. Vilar, A. J. deMello and J. C. deMello, *J. Mater. Chem.*, 2004, **14**, 2655–2660.
- [78] J. P. McMullen, M. T. Stone, S. L. Buchwald and K. F. Jensen, *Angew. Chem. Int. Ed.*, 2010, **49**, 7076–7080.
- [79] R. A. Bourne, R. A. Skilton, A. J. Parrott, D. J. Irvine and M. Poliakoff, *Org. Process Res. Dev.*, 2011, **15**, 932–938.
- [80] N. Holmes, G. R. Akien, A. J. Blacker, R. L. Woodward, R. E. Meadows and R. A. Bourne, *React. Chem. Eng.*, 2016, **1**, 366–371.
- [81] J. P. McMullen and K. F. Jensen, *Org. Process Res. Dev.*, 2010, **14**, 1169–1176.
- [82] B. J. Reizman and K. F. Jensen, *Chem. Commun.*, 2015, **51**, 13290–13293.
- [83] Z. Amara, E. S. Streng, R. A. Skilton, J. Jin, M. W. George and M. Poliakoff, *Eur. J. Org. Chem.*, 2015, **2015**, 6141–6145.
- [84] A. J. Parrott, R. A. Bourne, G. R. Akien, D. J. Irvine and M. Poliakoff, *Angew. Chem. Int. Ed.*, 2011, **50**, 3788–3792.
- [85] A. E. Kirby and A. R. Wheeler, *Lab Chip*, 2013, **13**, 2533–2540.
- [86] D. L. Browne, S. Wright, B. J. Deadman, S. Dunnage, I. R. Baxendale, R. M. Turner and S. V. Ley, *Rapid Commun. Mass Spectrom.*, 2012, **26**, 1999–2010.
- [87] N. Holmes, G. R. Akien, R. J. D. Savage, C. Stanetty, I. R. Baxendale, A. J. Blacker, B. A. Taylor, R. L. Woodward, R. E. Meadows and R. A. Bourne, *React. Chem. Eng.*, 2016, **1**, 96–100.
- [88] D. N. Jumbam, R. A. Skilton, A. J. Parrott, R. A. Bourne and M. Poliakoff, *J. Flow Chem.*, 2012, **2**, 24–27.
- [89] J. S. Moore and K. F. Jensen, *Org. Process Res. Dev.*, 2012, **16**, 1409–1415.
- [90] S. Krishnadasan, A. Yashina, A. J. deMello and J. C. deMello, in *Adv. Chem. Eng.*, ed. J. Schouten, Elsevier, 2010, vol. 38, ch. 4, pp. 195–231.
- [91] J. R. S. C. Mateo, in *Weighted Sum Method and Weighted Product Method*, Springer, London, 2012, vol. 83, ch. 4, pp. 19–22.
- [92] R. T. Marler and J. S. Arora, *Struct. Multidiscip. Optim.*, 2010, **41**, 853–862.
- [93] E. Triantaphyllou and B. Shu, *Encycl. Electr. Electron. Eng.*, 1998, **15**, 175–186.
- [94] D. Cortés-Borda, K. V. Kutonova, C. Jamet, M. E. Trusova, F. Zammattio, C. Truchet, M. Rodriguez-Zubiri and F. X. Felpin, *Org. Process Res. Dev.*, 2016, **20**, 1979–1987.

- [95] W. Huyer and A. Neumaier, *ACM Trans. Math. Softw.*, 2008, **35**, 1–25.
- [96] K. M. Tibbetts, X.-J. Feng and H. Rabitz, *Phys. Chem. Chem. Phys.*, 2017, **19**, 4266–4287.
- [97] R. M. Myers, D. E. Fitzpatrick, R. M. Turner and S. V. Ley, *Chem. Eur. J.*, 2014, **20**, 12348–12366.
- [98] M. D. Hoey and D. C. Dittmer, *J. Org. Chem.*, 1991, **56**, 1947–1948.
- [99] K. H. Kim, H. Kang, S. Y. Nam, J. Jung, P. S. Kim, C. H. Cho, C. Lee, S. C. Yoon and B. J. Kim, *Chem. Mater.*, 2011, **23**, 5090–5095.
- [100] W. H. Pirkle and M. S. Hoekstra, *J. Am. Chem. Soc.*, 1976, **98**, 1832–1839.
- [101] X. Meng, G. Zhao, Q. Xu, Z. Tan, Z. Zhang, L. Jiang, C. Shu, C. Wang and Y. Li, *Adv. Funct. Mater.*, 2014, **24**, 158–163.
- [102] Z. Li, H. Chen, H. Bao and M. Gao, *Chem. Mater.*, 2004, **16**, 1391–1393.
- [103] S. Sun and H. Zeng, *J. Am. Chem. Soc.*, 2002, **124**, 8204–8205.
- [104] D. J. Kang, H. Kang, C. Cho, K.-H. Kim, S. Jeong, J.-Y. Lee and B. J. Kim, *Nanoscale*, 2013, **5**, 1858–1863.
- [105] X. Meng, W. Zhang, Z. Tan, C. Du, C. Li, Z. Bo, Y. Li, X. Yang, M. Zhen, F. Jiang, J. Zheng, T. Wang, L. Jiang, C. Shu and C. Wang, *Chem. Commun.*, 2012, **48**, 425–427.
- [106] H. Kang, K. H. Kim, T. E. Kang, C. H. Cho, S. Park, S. C. Yoon and B. J. Kim, *ACS Appl. Mater. Interfaces*, 2013, **5**, 4401–4408.
- [107] D. E. Fitzpatrick, C. Battilocchio and S. V. Ley, *Org. Process Res. Dev.*, 2016, **20**, 386–394.
- [108] S. Dallwig, A. Neumaier and H. Schichl, in *GLOPT — A Program for Constrained Global Optimization*, ed. I. M. Bomze, T. Csendes, R. Horst and P. M. Pardalos, Springer, Boston, MA, 1997, pp. 19–36.
- [109] L. Wu, A. Mendoza-Garcia, Q. Li and S. Sun, *Chem. Rev.*, 2016, **116**, 10473–10512.
- [110] M. P. Morales, S. Veintemillas-Verdaguer, M. I. Montero, C. J. Serna, A. Roig, L. Casas, B. Martínez and F. Sandiumenge, *Chem. Mater.*, 1999, **11**, 3058–3064.
- [111] H. B. Na, I. C. Song and T. Hyeon, *Adv. Mater.*, 2009, **21**, 2133–2148.
- [112] N. Lee, D. Yoo, D. Ling, M. H. Cho, T. Hyeon and J. Cheon, *Chem. Rev.*, 2015, **115**, 10637–10689.
- [113] S. Laurent, D. Forge, M. Port, A. Roch, C. Robic, L. Vander Elst and R. N. Muller, *Chem. Rev.*, 2010, **110**, 2064–2110.
- [114] D. Wang, J. He, N. Rosenzweig and Z. Rosenzweig, *Nano Lett.*, 2004, **4**, 409–413.

- [115] L. H. Reddy, J. L. Arias, J. Nicolas and P. Couvreur, *Chem. Rev.*, 2012, **112**, 5818–5878.
- [116] C. A. Quinto, P. Mohindra, S. Tong and G. Bao, *Nanoscale*, 2015, **7**, 12728–12736.
- [117] D. Ling, W. Park, S. J. Park, Y. Lu, K. S. Kim, M. J. Hackett, B. H. Kim, H. Yim, Y. S. Jeon, K. Na and T. Hyeon, *J. Am. Chem. Soc.*, 2014, **136**, 5647–5655.
- [118] Y. W. Jun, J. W. Seo and J. Cheon, *Acc. Chem. Res.*, 2008, **41**, 179–189.
- [119] H. Gu, P. L. Ho, K. W. Tsang, L. Wang and B. Xu, *J. Am. Chem. Soc.*, 2003, **125**, 15702–15703.
- [120] A. Espinosa, R. Di Corato, J. Kolosnjaj-Tabi, P. Flaud, T. Pellegrino and C. Wilhelm, *ACS Nano*, 2016, **10**, 2436–2446.
- [121] P. Moroz, S. K. Jones and B. N. Gray, *Int. J. Hyperth.*, 2002, **18**, 267–284.
- [122] M. Gonzales-Weimuller, M. Zeisberger and K. M. Krishnan, *J. Magn. Magn. Mater.*, 2015, **321**, 1947–1950.
- [123] N. A. Frey, S. Peng, K. Cheng and S. Sun, *Chem. Soc. Rev.*, 2009, **38**, 2532–2542.
- [124] C. Sun, J. Lee and M. Zhang, *Adv. Drug Deliv. Rev.*, 2008, **60**, 1252–1265.
- [125] D. K. Yi, S. S. Lee and J. Y. Ying, *Chem. Mater.*, 2006, **18**, 2459–2461.
- [126] B. Movassagh and A. Yousefi, *Monatsh. Chem.*, 2015, **146**, 135–142.
- [127] A. M. Abu-Dief and S. M. Abdel-Fatah, *Beni-Suef Univ. J. Basic Appl. Sci.*, 2017.
- [128] M. I. Khalil, *Arab. J. Chem.*, 2015, **8**, 279–284.
- [129] Z. Li, B. Tan, M. Allix, A. I. Cooper and M. J. Rosseinsky, *Small*, 2008, **4**, 231–239.
- [130] S. J. Lee, J. R. Jeong, S. C. Shin, J. C. Kim and J. D. Kim, *J. Magn. Magn. Mater.*, 2004, **282**, 147–150.
- [131] S. Yu and G. M. Chow, *J. Mater. Chem.*, 2004, **14**, 2781–2786.
- [132] W. W. Yu, J. C. Falkner, C. T. Yavuz and V. L. Colvin, *Chem. Commun.*, 2004, **0**, 2306–2307.
- [133] F. X. Redl, C. T. Black, G. C. Papaefthymiou, R. L. Sandstrom, M. Yin, H. Zeng, C. B. Murray and S. P. O’Brien, *J. Am. Chem. Soc.*, 2004, **126**, 14583–14599.
- [134] J. Park, K. An, Y. Hwang, J. E. Park, H. J. Noh, J. Y. Kim, J. H. Park, N. M. Hwang and T. Hyeon, *Nat. Mater.*, 2004, **3**, 891–895.
- [135] C. Feldmann and H. Jungk, *Angew. Chem. Int. Ed.*, 2001, **40**, 359–362.
- [136] J. Wang, J. Wan and K. Chen, *Mater. Lett.*, 2010, **64**, 2373–2375.
- [137] J. Park, E. Lee, N.-M. Hwang, M. Kang, S. C. Kim, Y. Hwang, J.-G. Park, H.-J. Noh, J.-Y. Kim, J.-H. Park and T. Hyeon, *Angew. Chem. Int. Ed.*, 2005, **44**, 2872–2877.

- [138] L. Frenz, A. El Harrak, M. Pauly, S. Bégin-Colin, A. D. Griffiths and J. C. Baret, *Angew. Chem. Int. Ed.*, 2008, **47**, 6817–6820.
- [139] A. Abou Hassan, O. Sandre, V. Cabuil and P. Tabeling, *Chem. Commun.*, 2008, **0**, 1783–1785.
- [140] W.-B. Lee, C.-H. Weng, F.-Y. Cheng, C.-S. Yeh, H.-Y. Lei and G.-B. Lee, *Biomed. Microdevices*, 2009, **11**, 161–171.
- [141] W. Glasgow, B. Fellows, B. Qi, T. Darroudi, C. Kitchens, L. Ye, T. M. Crawford and O. T. Mefford, *Particuology*, 2016, **26**, 47–53.
- [142] L. Zhang, G. Niu, N. Lu, J. Wang, L. Tong, L. Wang, M. J. Kim and Y. Xia, *Nano Lett.*, 2014, **14**, 6626–6631.
- [143] K. Kumar, *Ph.D. thesis*, Imperial College London, 2013.
- [144] S. K. Yap, W. K. Wong, N. X. Y. Ng and S. A. Khan, *Chem. Eng. Sci.*, 2017, **169**, 117–127.
- [145] L. Mazutis, J.-C. Baret, P. Treacy, Y. Skhiri, A. F. Araghi, M. Ryckelynck, V. Taly and A. D. Griffiths, *Lab Chip*, 2009, **9**, 2902–2908.
- [146] E. Um, D. S. Lee, H. B. Pyo and J. K. Park, *Microfluid. Nanofluidics*, 2008, **5**, 541–549.
- [147] Y. Wang, E. Tumarkin, D. Velasco, M. Abolhasani, W. Lau and E. Kumacheva, *Lab Chip*, 2013, **13**, 2547–2553.
- [148] H. C. Peng, Z. Li, G. Aldahondo, H. Huang and Y. Xia, *J. Phys. Chem. C*, 2016, **120**, 11754–11761.
- [149] H.-Y. Wu, H.-C. Chu, T.-J. Kuo, C.-L. Kuo and M. H. Huang, *Chem. Mater.*, 2005, **17**, 6447–6451.
- [150] Q. Zhang, W. Li, C. Moran, J. Zeng, J. Chen, L. P. Wen and Y. Xia, *J. Am. Chem. Soc.*, 2010, **132**, 11372–11378.
- [151] K. Manthiram, B. J. Beberwyck, D. V. Talapin and A. P. Alivisatos, *J. Vis. Exp.*, 2013, **82**, e50731–e50738.
- [152] B. J. Deadman, D. L. Browne, I. R. Baxendale and S. V. Ley, *Chem. Eng. Technol.*, 2015, **38**, 259–264.
- [153] Z. Xu, C. Shen, Y. Hou, H. Gao and S. Sun, *Chem. Mater.*, 2009, **21**, 1778–1780.
- [154] P. Guardia, A. Labarta and X. Batlle, *J. Phys. Chem. C*, 2011, **115**, 390–396.
- [155] B. Shahriari, K. Swersky, Z. Wang, R. P. Adams and N. De Freitas, *Proc. IEEE*, 2016, **104**, 148–175.