A FRAMEWORK FOR UNDERSTANDING AND CONTROLLING BATCH COOLING CRYSTALLIZATION

A Thesis
Presented to
The Academic Faculty

by

Daniel J. Griffin

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Chemical & Biomolecular Engineering

Georgia Institute of Technology
August 2016

Copyright © 2016 by Daniel J. Griffin

A Framework for Understanding and Controlling Batch Cooling Crystallization

Approved by:

Dr. Ronald W. Rousseau, Advisor
School of Chemical & Biomolecular Engineering
Georgia Institute of Technology

Dr. Martha A. Grover, co-Advisor
School of Chemical & Biomolecular Engineering
Georgia Institute of Technology

Dr. Yoshiaki Kawajiri. co-Advisor
School of Chemical & Biomolecular Engineering
Georgia Institute of Technology

Dr. Matthew J. Realff
School of Chemical & Biomolecular Engineering
Georgia Institute of Technology

Dr. Bojan Petrovic
School of Nuclear & Radiological Engineering
Georgia Institute of Technology

Date Approved: June 14, 2016

ACKNOWLEDGEMENTS

I am greatly indebted to my three Advisors at Georgia Tech, Professors Ronald Rousseau, Martha Grover and Yoshiaki Kawajiri. In working with Ron, Martha, and Yoshi I have enjoyed the application of a collective intellect that far exceeds my own; insofar as I have produced work that will persist beyond my PhD, it is to their credit. They have also made the work fun—I would happily spend five more years engaging in research and learning from each of them.

I am also indebted to two graduate students ahead of me: Huayu Li (now at AbbVie) and Luis Encarnacion-Gomez (now at Bristol-Myers Squibb). Not only did Huayu and Luis teach me about crystallization, they also demonstrated a rare quality that I have since tried (sometimes in vain) to copy: the ability to work very hard and set high goals, while maintaining a calm and cheerful attitude. Striving to work as Huayu and Luis did has helped me a great deal in graduate school.

In completing my PhD and writing this thesis, I feel a deep sense of gratitude to my family. At my best, I hope I am their reflection.

My parents are incredibly strong and have supported me every step of the way. My older brother Ryan is the sharpest person I know; I expect I have worked so hard in school, in part, because I have wanted to keep pace with Ryan's ability to think deeply about so many different topics. And Mike, my younger brother, is the best. He continues to show me that it *is* possible to be really great at all you do without the slightest hint of boastfulness or air of taking yourself too seriously.

I am also close with my extended family and there is someone who I am missing as I complete this chapter: my uncle Jay. To me, Uncle Jay was someone to emulate. He was the epitome of discipline, intelligence, and confidence. He was also playful and witty. I don't remember a time he wasn't grinning, waiting to say something wry when I greeted him and I can imagine his response now: "Spaniel! They passed *you*?" This thesis is dedicated to Jay, his playful spirit and his insistence that hard work is worth it in the end.

Finally, to my girlfriend, Liane: there is very little that can't be fixed by an evening run with you. In the past two and a half years you have been my counselor, adviser and best friend. And, in doing so, you have influenced my work and outlook more than you realize. I can't tell you the freedom and piece of mind that comes from finding a partner that I know will share and elevate any triumphs and diminish and wash away any failures. Thank you.

TABLE OF CONTENTS

PART I THEORY AND FRAMEWORKS

CONCLUSIONS AND FUTURE WORK

LIST OF TABLES

LIST OF FIGURES

SUMMARY

IN taking a different view of crystallization dynamics, this thesis reveals a new framework for addressing a prevalent process engineering challenge: control over the size of crystals produced by batch cooling crystallization.

The thesis divides roughly into halves. In the first half, the crystal size control problem is introduced and the proposed framework for addressing this problem—termed the mass-count (MC) framework—is developed. This new framework is laid out along side the population balance (PB) framework, which is the prevailing framework for modeling crystallization dynamics and addressing the crystal size control problem.

In putting the proposed and established frameworks side by side, the intent is not to say that one or the other is *correct*. Rather, the point is to show that they are different perspectives that facilitate different control approaches. The PB framework is built up from first principles; it is intellectually stimulating and mathematically complete, but it has a drawback for application: it does not directly enable feedback control. The MC framework, on the other hand, takes a less detailed view of crystallization dynamics and does not connect to crystallization theory as directly; it is also more conducive to application.

In the second half of the thesis, the utility of the MC framework is put to the test. The framework is first applied to understand and model the crystallization dynamics for two widely different systems: darapskite salt crystallization from water and paracetamol crystallization from ethanol. Once the dynamics have been modeled, the framework is then used to develop feedback control schemes. These schemes are applied to both experimental systems and, in both cases, crystal size control is demonstrated.

INTRODUCTION

BATCH cooling crystallization is widely applied to isolate high-purity chemicals and pharmaceuticals. The operation is straightforward in principle: the temperature of a solution is reduced to drive the crystallization of a target component, which can then be filtered and isolated. Detailed examination, however, reveals complexity. Crystals develop and evolve through a series of dynamic processes occurring at the molecular level. These dynamics determine the size, shape, and purity of the product crystals—leaving the process control engineer with the challenging task of figuring out how to manipulate the input variable, temperature, to influence the underlying crystallization dynamics and ultimately produce crystals with specific desired properties.

> *This thesis provides a useful framework for addressing such a task when the objective is to control the mean crystal size.*

Background on crystal size control. Crystal size control has taken a leading role in research on crystallization operations. This is justified, to an extent, by the importance of controlling the crystal size for both product quality and the performance of downstream filtration and solids-handling operations.

Work on crystal size control was recorded in academic journals as early as the 1920's [Griffiths, 1924, Montillon and Badger, 1927, McCabe, 1929], but largely gained prevalence in the 1960's when the population balance equation was introduced [Hulburt and Katz, 1964, Randolph, 1964]. The population balance equation provided a mathematical framework for modeling the evolution of a crystal population and established the primary paradigm from which crystal size control has been considered since.

Initially, the population balance framework was used to predict the size of crystals resulting from various process configurations under idealized steady-state operation—with emphasis placed on the mixed-suspension, mixed-product removal (MSMPR) configuration [Randolph and Larson, 1988]. By the 1990's, computing power had increased substantially and new measurement tools enabled online monitoring of key properties—notably, solution concentration [Dunuwila et al., 1994].

With these advancements, population balance models were extended to transient operations [Rawlings et al., 1993, Miller and Rawlings, 1994] and control strategies developed within the population balance framework became more applicable to industrial practice [Nagy and Braatz, 2012]. Although useful for designing the operation, these strategies were generally still implemented in an open-loop fashion and susceptible to inconsistent results in the face of model-mismatch and process uncertainty.

At the start of the 21st century, the U.S. Food and Drug Administration initiative promoted "innovation and efficiency in pharmaceutical manufacturing and quality assurance" through the use and real-time incorporation of online measurements [PAT, 2004, Yu et al., 2004, Simon et al., 2015]. This encouragement, along with ever-increasing innovation in population balance modeling and computation, has ushered in crystal size control strategies that are more robust and geared towards application [Braatz et al., 2002, Braatz, 2002, Worlitschek and Mazzotti, 2004, Barrett et al., 2005, Larsen et al., 2006, Corriou and Rohani, 2008, Nagy, 2009, Mesbah et al., 2010, Mesbah et al., 2012, Nagy and Braatz, 2012, Nagy et al., 2013, Acevedo et al., 2015]. Nevertheless, crystal size control in industry remains an outstanding challenge.

The point of this thesis. The persistence of the crystal size control problem speaks, in part, to the difficulty of the task. But I also see another impedance: the control problem is almost always approached from the same perspective—that established by the population balance framework—and this perspective is not particularly amenable to establishing control *in practice*.

The point of this thesis is to show that a different, less detailed, view of crystallization dynamics yields a new framework that lets us directly address the crystal size control problem.

OVERVIEW

This thesis is divided into two parts:

Part I—Theory and Frameworks—is made up of three chapters. Chapter 1 introduces key concepts of batch cooling crystallization and the crystal size control problem. Chapters 2 and 3 respectively present two frameworks for addressing the crystal size control problem: the established population balance (PB) framework is presented in Chapter 2, while the mass-count (MC) framework—central to this thesis—is developed in Chapter 3. In introducing the MC framework, a practical route for achieving crystal size control is stated. This is put to the test in Part II.

Part II—Application—is also constructed from three chapters. Chapter 4 gives two experimental crystallization systems and also describes the online measurement tools that are instrumental to applying the MC framework. Chapter 5 illustrates the application of the MC framework to understand and model the crystallization dynamics. Finally, Chapter 6 gives control results. The results show that—using the MC framework—we can achieve crystal size control in practice.

A Framework for Understanding and Controlling Batch Cooling Crystallization

PART I

Theory and Frameworks

CHAPTER 1

BATCH COOLING CRYSTALLIZATION:

KEY CONCEPTS AND THE CRYSTAL SIZE CONTROL PROBLEM

1.1 States of Matter and the Driving Force for Crystallization

Collections of molecules exist in different *states of matter* or *phases.* In a *gaseous* state, molecular density is low and the molecules are in constant random motion. In a *liquid* state, the molecules are packed closer together and their motion is fluid. In a *solid* state, the molecules are constrained, oscillating around fixed positions—if these positions are arranged in a regular pattern, known as a lattice, the solid state is said to be crystalline.

Thermodynamic variables dictate the equilibrium state. The equilibrium or favored state of a collection of molecules is a function of the macroscopic thermodynamic variables, temperature and pressure. As a consequence, we can often predictably manipulate the phase by changing these macroscopic variables. The effect of temperature on the state of water, for example, is ubiquitous and commonly observed: solid states are favored at temperatures below 0 °C and gaseous states are favored at temperatures above 100 °C (at atmospheric pressure).

The story is slightly more involved for solutions, which contain multiple different types of molecules; for solutions the equilibrium state at a given temperature and pressure may be multiphasic—having, for example, portions of the system in a solid state and portions in a liquid state.

Free energy. The concept of free energy is useful for describing the evolution of a system towards equilibrium. A system in a given state has associated with it an amount of energy “free for work”. The equilibrium state, or favored state, for a system under set thermodynamic conditions has minimal free energy. Moreover, the rate at which a system moves towards the equilibrium state appears to depend on the magnitude of the reduction of free energy achieved. In this sense, the evolution of a system towards the equilibrium state is *driven* by the reduction of free energy.

Batch cooling crystallization. Batch cooling crystallization operations start with a liquid solution. Reducing the temperature changes the equilibrium, minimum-energy state of the system—at some point favoring the formation of a solid crystalline phase made of one or more of the solution components. Crystallization is the dynamic process by which the new solid crystalline phase forms from the liquid solution.

Thermodynamic driving force for crystallization. For crystallization, it is convenient to express free energy on a per-molecule basis. That is, in terms of chemical potential, μ: the change in free energy associated with the addition or subtraction of a single molecule.

The thermodynamic driving force for crystallization can then be expressed. It is the difference in chemical potential of the crystallizing component, i, in solution and the chemical potential of that same component, i, in a pure crystalline state at the same temperature and pressure:

$$\Delta\mu^{\mathrm{cryst}} \equiv \mu_i^{\mathbf{(L)}}(\mathbf{C}, T, P) - \mu_i^{\mathbf{(S)}}(T, P), \tag{1.1}$$

where $\mu_i^{\mathbf{(L)}}$ is the chemical potential of component i in a solution of composition $\mathbf{C}$ at temperature T and pressure P; $\mu_i^{\mathbf{(S)}}$ is the chemical potential of i in a pure crystalline state under the same temperature and pressure. The relevant states are illustrated in Figure 1.1.

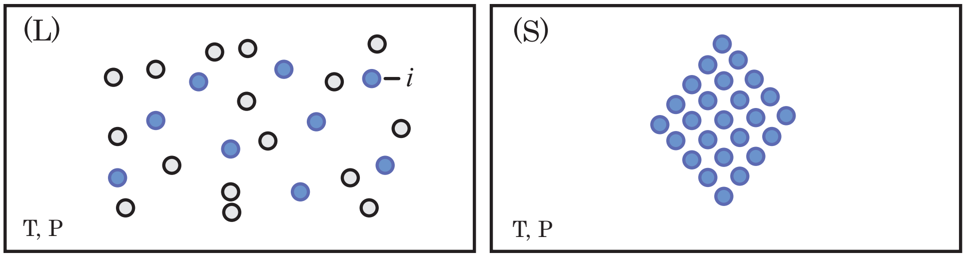

Figure 1.1: Relevant states for expressing the thermodynamic driving force for crystallization. *Left*: a solution containing component i in a liquid state at temperature T and pressure P. *Right*: pure component i in a solid crystalline state at the same temperature T and pressure P.

Solubility. For a multiphasic system that contains both a liquid solution phase and a solid phase composed of pure component i (termed the solute), the concentration of solute (i) in solution at *equilibrium* under a given temperature and pressure is the *solubility* of that component in the given solution:

$$\mu_i^{\mathbf{(L)}}(C_i^*, T, P) = \mu_i^{\mathbf{(S)}}(T, P), \tag{1.2}$$

where C^* is the solubility of component i at the given temperature T and pressure P.

Solubility curve. As the pressure has only a minor effect, solubility is usually expressed in terms of the equilibrium concentration of the solute in solution as a function of temperature: $C_i^* = C_i^*(T)$. This is the solubility curve.

Figure 1.2 shows a typical solubility curve. There is an important attribute of this curve: the slope of the solubility-temperature relationship is positive. This attribute is a prerequisite for running a batch *cooling* crystallization operation—only when the solubility-temperature relationship is positive, will cooling create a driving force for crystallization.

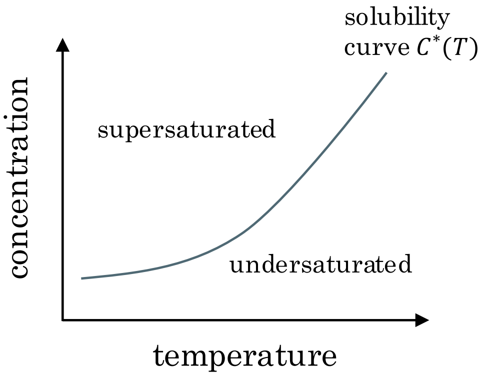

Figure 1.2: Illustration of a typical solubility curve.

Supersaturation. A solution is said to be supersaturated when the concentration of the crystallizing component, C_i, is greater than the solubility concentration, C_i^*, for the given temperature.

There are a number of measures of supersaturation, the most popular are simply the concentration–solubility difference and the normalized concentration–solubility difference:

$$\Delta C_i \equiv C_i - C_i^*(T), \text{ and} \tag{1.3}$$

$$\sigma_i \equiv \frac{C_i - C_i^*(T)}{C_i^*(T)}. \tag{1.4}$$

In this thesis, ΔC will be referred to as the concentration driving force and "supersaturation" will be used in reference to σ.

Supersaturation is a measure of the thermodynamic driving force for crystallization. Supersaturation, σ, is related to the thermodynamic driving force for crystallization, μ^{cryst}. Under ideal solution thermodynamics, for example, it can be shown that $\mu^{\text{cryst}} \propto T\sigma$, where T is the absolute solution temperature.

More generally, σ provides a measure that at least gets the sign right: when a solution is supersaturated, there exists a driving force for crystallization; when a solution is undersaturated, there exists a driving force for dissolution. Mathematically:

$$\mu^{\text{cryst}} > 0 \iff \sigma > 0. \tag{1.5}$$

Temperature changes can be used to manipulate σ and thereby influence the crystallization dynamics. When the solubility curve is positive, temperature reductions create supersaturation and a driving force for crystallization. Moreover, the magnitude of a temperature change dictates the degree of supersaturation achieved, which in turn affects the crystallization dynamics. These dynamics, and the effect of supersaturation on them, will be discussed in more detail in the following section.

1.2 Crystallization Dynamics

Crystallization occurs by nucleation and growth of individual crystals. Once formed, crystals can subsequently dissolve, agglomerate, and break apart. These processes, illustrated in Figure 1.3, collectively constitute crystallization dynamics.

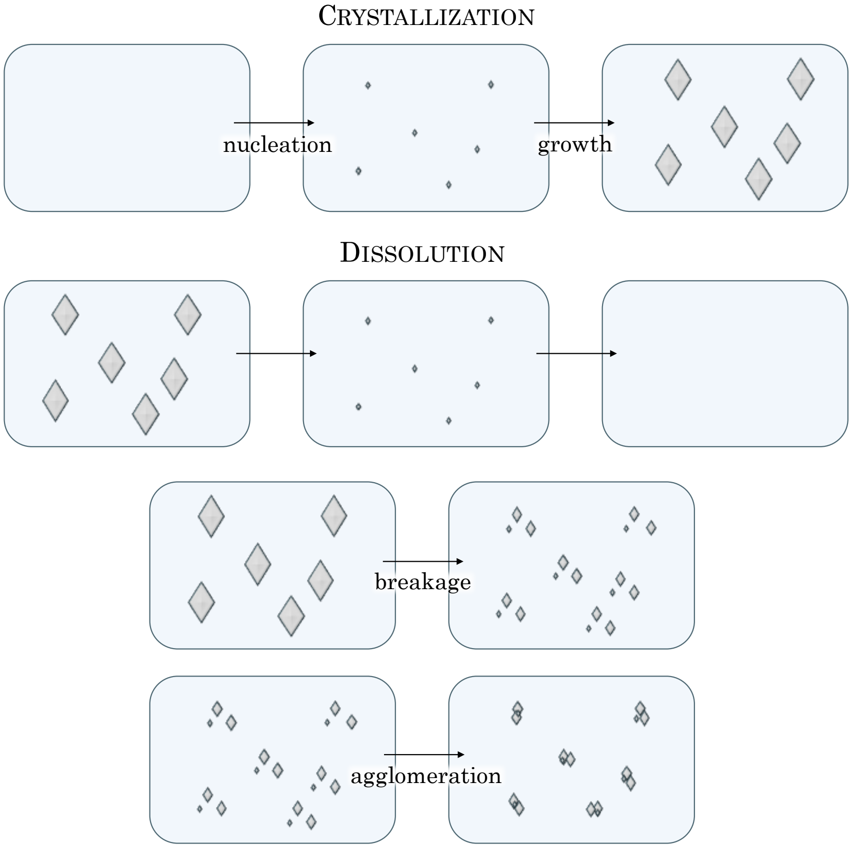

Figure 1.3: Illustration of the underlying mechanisms by which crystals develop and evolve.

1.2.1 Nucleation and crystal growth

When a solution is supersaturated there exists a driving force for crystallization. Another way of looking at this: supersaturation provides a driving force for the net transfer of mass from liquid phase to solid crystalline phase, which occurs through nucleation of new crystals and growth of existing crystals.

Nucleation. Nucleation refers to the initial formation of a new phase from an existing one. In context: the formation of a new crystal from solution. Mullin [Mullin, 2001] suggests categorizing nucleation as shown in Figure 1.4. This leaves three different categories: homogeneous nucleation, heterogeneous nucleation, and secondary nucleation.

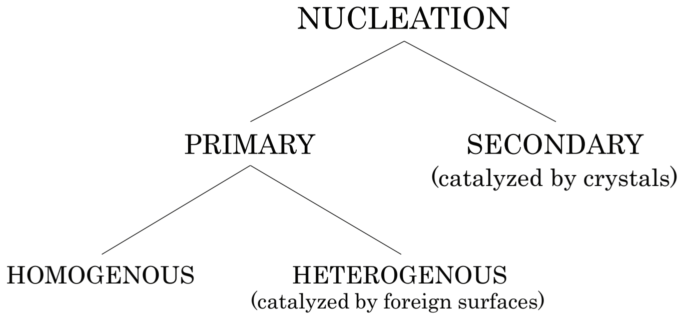

Figure 1.4: Categorization of nucleation [Mullin, 2001].

Homogeneous nucleation. Homogeneous nucleation refers to the formation of a new crystal nuclei from an evenly-mixed solution. Exactly how this occurs—that is, how the solute elements aggregate in solution and arrange into a crystal lattice—is not clear. In fact, there is substantial evidence of multiple pathways by which homogeneous nucleation may occur, even for the same system [Erdemir et al., 2009, Vekilov, 2010, Baumgartner et al., 2014, Erdemir et al., 2013, Nielsen et al., 2014].

Figure 1.5 illustrates two potential homogeneous nucleation pathways. In the first, molecules aggregate and order at the same time; in the second, molecules first aggregate and then order.

The pathway has an important, albeit nebulous, effect on the rate or likelihood of nucleation. Nucleation is driven by solution thermodynamics and expected to occur when the solution is supersaturated, but the rate of nucleation also depends on a surface energy barrier to forming a new phase [Gibbs, 1876, Gibbs, 1878], which in turn depends on the pathway [Vekilov, 2010].

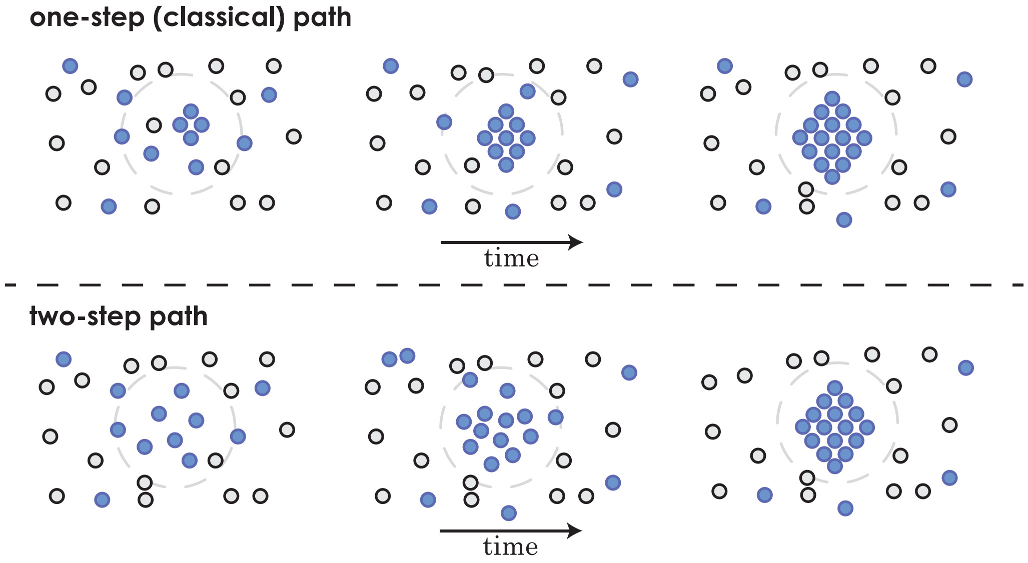

Figure 1.5: Different homogeneous nucleation pathways: in the classical mechanism, molecules condense and order simultaneously; this is opposed to the two-step mechanism in which the molecules condense first and then order.

Heterogeneous nucleation. Heterogeneous nucleation refers to nucleation catalyzed by the presence of a foreign surface (due to the vessel walls, mixing propeller, dust particles, etc.). These foreign surfaces are expected to reduce the surface energy barrier to nucleation, and thereby increase the rate of nucleation.

Secondary nucleation. Secondary nucleation refers to nucleation catalyzed by the existence already-formed crystals. This catalysis may be caused simply by the presence of the crystal surface, which reduces the energy barrier to nucleation, or by a variety of *breeding* mechanisms in which small crystal fragments dislodge from the existing crystals and serve as seeds for new nuclei.

Growth. Once stable crystal nuclei have formed, they may grow. Crystal growth is often considered to consists of two steps: diffusion of the solute molecule from the bulk solution to the surface of a crystal and incorporation of that solute molecule into the crystal. Depending on the conditions, one of these steps may be rate limiting or both steps may contribute. Lewis, Seckler, Kramer, and van Rosmalen [Lewis et al., 2015] suggests categorizing crystal growth according to which step dominates the rate of growth as shown in Figure 1.6.

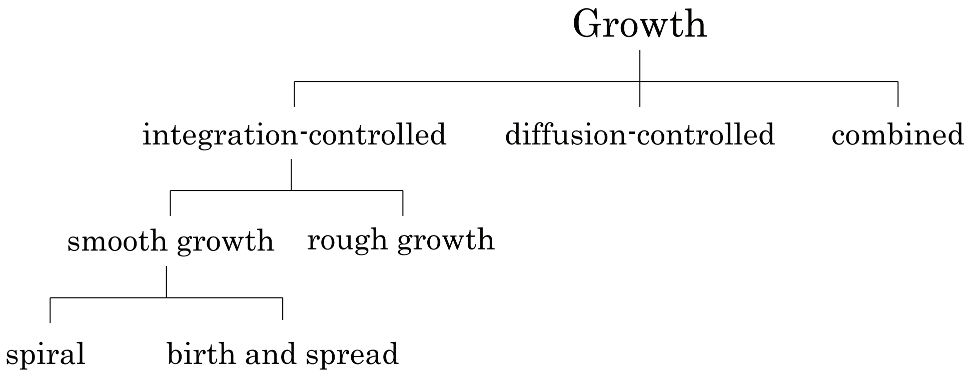

Figure 1.6: Categorization of growth by the rate-controlling mechanism [Lewis et al., 2015].

Integration-controlled growth. When integration is the rate limiting step, growth may be sub-categorized according to the surface-incorporation mechanism. There are three prominent incorporation mechanisms suggested by [Lewis et al., 2015] that lead to different integration-controlled growth categories: rough growth, spiral growth, and birth-and-spread growth.

Rough growth is favored under high supersaturation and is characterized by the rapid deposition of molecules onto the surface of an existing crystal. The placement of these molecules in largely random. This is in contrast to spiral growth and birth-and-spread growth, in which deposit at propagating ‘steps’ on the crystal surface. In spiral growth, new molecules deposit on step edges that propagate in a spiral pattern; this type of growth is generally slow and observed under low supersaturation. In birth-and-spread growth, molecules deposit on step edges that spread out radially from one or more surface nuclei; this type of growth is generally observed under moderate levels of supersaturation and occurs at more moderate rates.

Diffusion-controlled growth. Under stagnant conditions, the incorporation of molecules onto a crystal surface may occur more rapidly than new molecules diffuse from the bulk solution. In such cases, the rate of growth depends primarily on the concentration gradient (between the bulk fluid and the local solution region near the crystal surface) and the diffusion characteristics of the solution.

Combined growth. In other cases, both the integration and diffusion steps may contribute to the rate of growth. In such cases, the growth rate expressions are more difficult to pin down and are generally taken to be empirical functions of the supersaturation.

Size-dependent growth and growth rate dispersion. In discussing the categories of growth given by Figure 1.6 it is tacitly assumed that grow rates depend on the solution properties and not the characteristics of the crystals themselves. Empirical observations, however, suggests that the crystal characteristics also influence the growth rate. The growth rate, for example, may be *size-dependent*: with the size of the crystal influencing the rate at which it grows. It can also depend on more intricate details of the crystal that are harder to observe or control, such as degree of disorder in the crystal lattice. This is supported by the observation that crystals of the same size under the same solutions conditions, grow at varying rates—a phenomena known as *growth-rate dispersion* [White and Write, 1971, Zumstein and Rousseau, 1987a, Zumstein and Rousseau, 1987b].

1.2.2 Dissolution

When a crystal-containing solution is undersaturated there exists a driving force for the net transfer of mass from the solid to the liquid phase. This occurs through dissolution.

Dissolution, here, refers to the opposite of crystallization. This process, like crystallization, is driven by thermodynamics: when a solution is undersaturated, there exists a driving force for dissolution. The key difference in the dynamics of dissolution, as compared to the exact mirror opposite of crystallization, is the absence of a surface energy barrier in dissolution. Lacking an energy barrier, the rate of dissolution is generally rapid, even at low levels of undersaturation.

1.2.3 Agglomeration and breakage

The dynamic mechanism discussed so far—crystallization and dissolution—cause a change in the distribution of mass between the crystal phase and liquid phase: crystallization occur when the solution is supersaturated and causes a net shift in mass from the liquid phase to the crystal phase; dissolution, on the other hand, occurs when the solution is undersaturated and causes a net shift in mass from the crystal phase to the solution phase. But the characteristics of crystals may also change without a net transfer of mass between phases. Two ways in which this can happen are crystal agglomeration and crystal breakage.

Agglomeration. Crystal agglomeration is the process in which individual crystals that have already formed in solution come in contact and fuse together. This dynamic process can be complex, with the rate or likelihood of agglomeration being a multivariate function that depends on, at least: the solution agitation and flow fields; the crystal number density; the size of the crystals; the inter-crystal forces; and the thermodynamics of crystal bridge formation.

Breakage. Crystal breakage is in many ways the opposite of agglomeration. It is the division of a current single crystals into two or more crystal fragments. This process is largely independent of the solution thermodynamic properties. Instead, the rate or likelihood of crystal breakage depends heavily on the physical properties of the crystals and the kinetic and physical environment: how fragile the crystal are and how vigorously the solution is stirred.

1.3 The Crystal Size Control Problem (for unseeded batch cooling crystallization)

The underlying dynamic processes by which crystals form and evolve—nucleation, growth, dissolution, agglomeration, and breakage—collectively govern the number of crystals and the size of those crystals. Crystal size control, then, comes down to expertly inducing or suppressing these underlying processes and also tailoring the relative rates at which they occur.

This can potentially be achieved through manipulation of supersaturation enacted via temperature changes. But, as the previous section is intended to convey, the underlying dynamic processes are complex and may depend on more than just the solution thermodynamics. Moreover, multiple underlying processes, nucleation and growth for example, may occur simultaneously. These complications convolute the relationship between the process input—the temperature profile—and the control variable—the mean size of the produced crystals. As such, the crystal size control problem, illustrated in Figure 1.7, can be one of considerable difficulty.

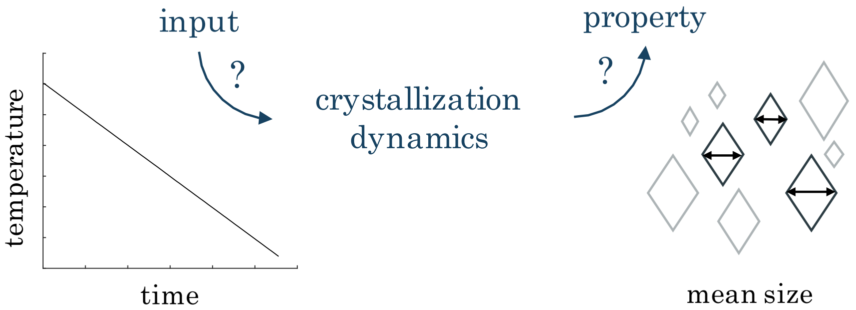

Figure 1.7: The crystal size control problem for unseeded batch cooling crystallization.

A framework is needed to establish crystal size control. To address the crystal size control problem, a framework is needed for modeling crystallization dynamics and establishing the essential relationship between the input temperature manipulations and the final control property—the crystal size. Two such frameworks are described in the following two chapters: the first is the well-established population balance (PB) framework; the second is the mass-count (MC) framework that is central to this thesis. A key advantage of the MC framework, as will be described, is that it is designed to directly facilitate feedback control.

CHAPTER 2

THE POPULATION BALANCE (PB) FRAMEWORK

MODELING a dynamical system starts with characterizing the state of the system at any given point in time. For crystallization, this means finding a way to characterize both the state of the solution as well as the state of the dispersed crystals.

Provided that the solution is well-mixed and the pressure is fixed, it is straightforward to describe the solution state: the solution state is given by the composition and temperature. Characterizing the state of a dispersed crystal population is more difficult and represents more of an open choice.

As described in this chapter, we might characterize the crystal state by a crystal size distribution density function, $n(L, t)$. This does not fully describe the crystal state in the sense that it contains *all* of the information about the crystals and perfectly represents reality. It does, however, capture the size of the crystals—the property we aim to control—and it is also convenient because it yields a continuity equation for the rate of change of the state: the population balance equation.

This strategy—characterizing the dispersed crystal state by the size distribution and writing a population balance expression for the rate of change of the state—is well-established and represents the orthodox framework for modeling crystallization dynamics and approaching the crystal size control problem.

2.1 Describing the Crystal State by the Crystal Size Distribution

A full description of a population of crystals dispersed in solution would include the position and velocity of each crystal—the so-called *external* properties—and also the size, shape, purity, and form of each crystal—the *internal* properties. Using such a detailed description, however, would be overkill. Simplifications are required to obtain a usable model.

In most applications, the crystal slurry is expected to be well mixed and the spatial distribution of the crystals is therefore of little interest. Moreover, we are often concerned with just one internal property—in this case, the crystal size.

When crystal size is of primary interest and the size of each crystal can be characterized by a single length, L, the crystal state may be described more simply in terms of the number of crystals and the size of each crystal. Figure 2.1 illustrates the evolution of a crystallization system with the crystal state described in this way.

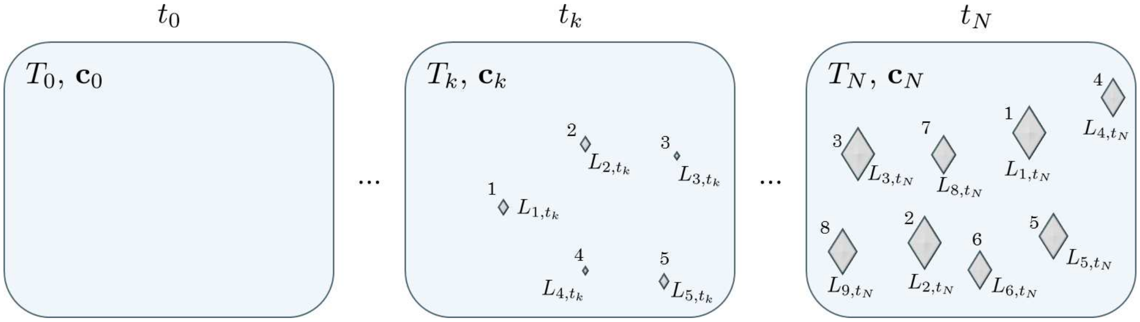

Figure 2.1: *Characterizing the state of a crystallization system.* At each instant in time the solution state is indicated by the solution temperature, T, and composition, $\mathbf{c}$; the crystal state is indicated by the number of crystals and the size of each crystal, L.

This representation has an associated numerical description: the dispersed crystal state at each instant can be captured by a vector, $\vec{L}$, with elements containing the length of each crystal. This is illustrated in Figure 2.2.

With this representation, we can imagine a numerical scheme that models the evolution of the crystal state: marching forward in time the nucleation of crystals can be represented by the addition of new elements to the vector $\vec{L}$ containing the size L° (assumed size of new nuclei), and the growth of crystals can be represented by an increase in the sizes, L_i's, listed in the existing elements. This numerical

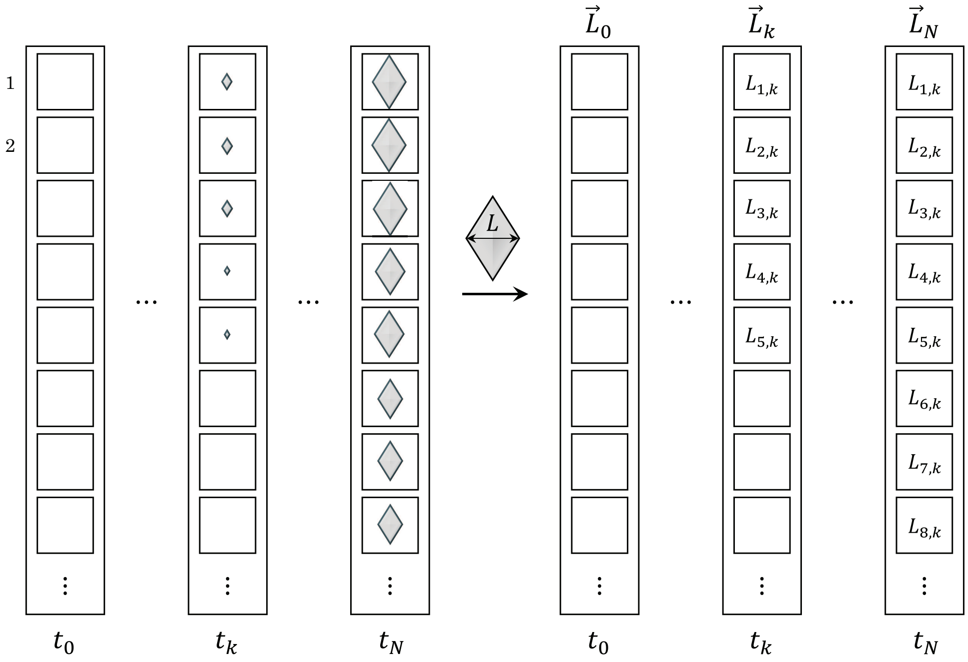

Figure 2.2: Evolution of the crystal state in terms of the number of crystals and the size of each of those crystals.

description of crystallization connects with the physical picture, but it also lacks an easy mathematical description and is an inefficient numerical scheme for large numbers of crystals.

There is another perspective that admits a compact mathematical representation and a more elegant mathematical model of the dynamics: the crystal state can represented by a crystal size distribution density function.

The crystal size distribution density function. Figure 2.3 displays information in the vector $\vec{L}$ as a histogram that shows the number of crystals that fall into bins of different size ranges. In showing the information as a histogram, we loose track of the individual crystals. That is, we no longer distinguish between crystal 1 and crystal 10 if they are of the same size. But we still keep track of the important information, which is the number of crystals and how these crystals are distributed according to size.

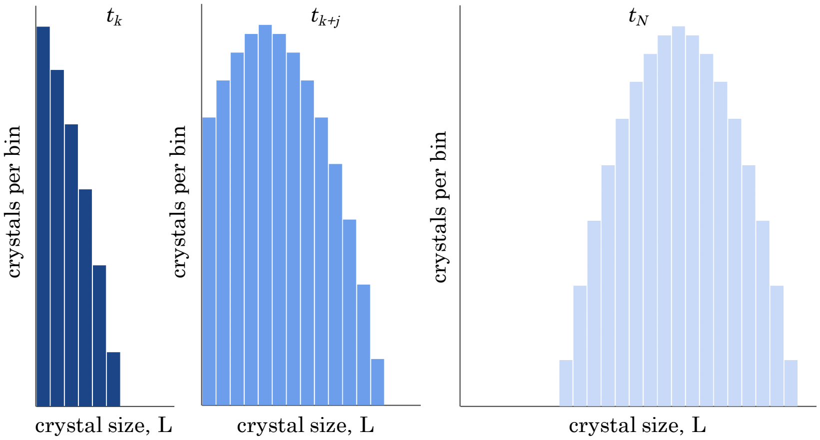

Figure 2.3: The crystal state, as represented by a size distribution histogram, at different times during a hypothetical crystallization process.

We can go a step further in making the representation compact and mathematically convenient. Dividing the number of crystals in each size bin by the width of that bin and treating this as a continuous variable, we arrive at a density function description of the crystal state:

$$n(L,t),\ \ L \in \mathbf{R}_+,\ \ t \in \mathbf{R}_+,$$

where $n(L,t)dL$ denotes the number of crystals with a size between L and $L + dL$. This crystal state representation is illustrated in Figure 2.4.

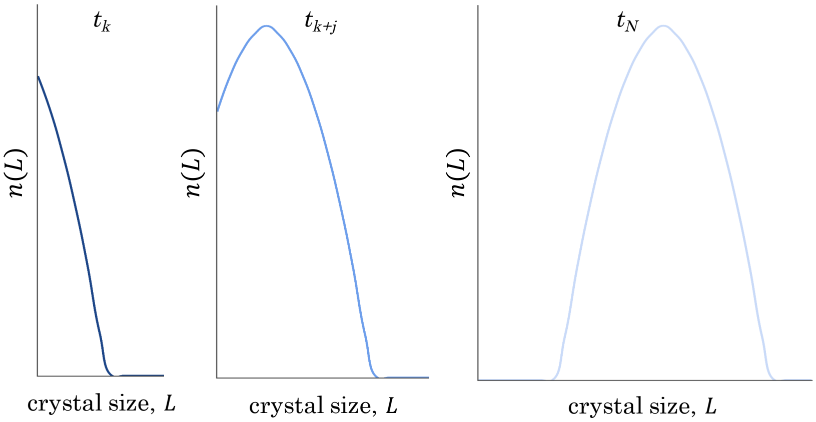

Figure 2.4: The crystal state, as represented by a size distribution density function, at different times during a hypothetical crystallization process.

2.2 Crystallization Dynamics Described by the Population Balance Equation

Having chosen to represent the crystal state with a crystal size distribution density function, the question is: how can we model the dynamics? That is, how can we mathematically express the rate of change of the crystal state as characterized by the size distribution density function?

Let me go back to the histogram representation of the crystal size distribution for a moment. Taking a single bin, we can write a balance expression for the rate of change of the number of crystals in the bin. The rate of change of the number of crystals in the bin must be equal to the flux *in*, minus the flux *out*. Assuming growth-only: the flux in equals the rate at which smaller crystals grow into the size range, and the flux out equals the rate at which crystals already in the size range grow to larger sizes. If other mechanisms are in play, such as agglomeration, the expressions for flux in and out become more complicated, but we can nevertheless write a balance on the rate of change of the number of crystals in the bin.

Here we see why a continuous crystal size distribution density function is a convenient state representation: under this representation, we can write a *continuity* equation for the rate of change of the state [Hulburt and Katz, 1964, Randolph, 1964, Ramkrishna, 2000]—much as we would do to describe the rate of change of mass or momentum in a given region in space [Bird et al., 2007].

Under the assumptions that $n(L,t)$ is continuously differentiable, the volume of solution is fixed, and the crystals change size by growth only (and do not dissolve, shrink, agglomerate, or break apart), the following continuity equation must hold:

$$\frac{\partial n(L,t)}{\partial t} + \frac{\partial\big(Gn(L,t)\big)}{\partial L} = 0. \tag{2.1}$$

This is the so-called, growth-only 1D *population balance equation*, where $G \equiv \partial L/\partial t$ is the crystal growth rate.

To be used, Equation 2.1 must be supplemented with a boundary condition:

$$n(0,t) = J/G, \tag{2.2}$$

where J is the nucleation rate giving the number of new nuclei formed per unit time; and an initial condition:

$$n(L,0) = 0.$$

(Here we assumed that no crystals are present initially).

The population balance equation given by (2.1) is the most common starting point for modeling crystallization dynamics and addressing the crystal size control problem.

Mass balance: connecting the solution state and the crystal state. A mass balance is used to connect the rate of change of the solution composition to the crystallization dynamics. For an unseeded batch cooling crystallization operation with a single-component solute, the mass balance can be expressed as follows:

$$C(t) = C(0) - \frac{1}{V} \underbrace{\rho k_v \int_0^\infty n(L,t) L^3 dL}_{\text{total mass of crystals}}, \tag{2.3}$$

where $C(t)$ is the concentration of the solute at time t, $C(0)$ is the initial composition, V is the solution volume, ρ is the crystal density, and k_v is a shape factor.

2.2.1 Completing a population balance *model*.

The population balance equation is incomplete as a model of crystallization dynamics. To fill in the model, expressions for the nucleation and growth rates must be specified. (For more sophisticated population balance expressions that include other mechanisms such as dissolution or agglomeration, the rates of these processes must also be specified).

Rate of nucleation as a function of σ and T. The simplest and most widely used theory that describes the rate of nucleation as function of the solution thermodynamic variables comes from so-called classical nucleation theory (CNT). The theory predicts the nucleation rate from a hypothetical free energy path originally derived for the condensation of vapor by Gibbs in 1876 [Gibbs, 1876, Gibbs, 1878]. This line of reasoning leads to an expression for the rate of nucleation as function of the solution temperature and supersaturation:

$$J = J_0 \exp(-\frac{b_0}{T^3\sigma^2}), \tag{2.4}$$

where J_0 and b_0 are constants.

Equation (2.4) makes the most sense for homogeneous nucleation occurring in a one-step manner; nucleation by other paths should have different free energy landscapes and, consequently, different rates. The presence of a foreign surface, for example, is expected to reduce the energy barrier to crystallization. Even more so if the surface is due to another crystal or if the nuclei forms from an existing crystal fragment. Heterogeneous primary nucleation and secondary nucleation are therefore expected to occur at a faster rate under the same thermodynamic conditions.

For different nucleation pathways, different rate functions have been proposed. Some have been developed from theoretical arguments, such as a rate for homogeneous nucleation occurring by a two-step mechanism [Vekilov, 2010], but the majority take an empirical form. In general, the rate of nucleation from solution is given as an increasing function of supersaturation, temperature, and the number of crystals already present in solution:

$$J = \underbrace{J_{0,1}\exp(-E_1/T)\sigma^{b_1}}_{\text{primary nucleation}} + \underbrace{J_{0,2}\exp(-E_2/T)\sigma^{b_2}A_{\mathrm{C}}/V_{\mathrm{S}}}_{\text{secondary nucleation}}, \tag{2.5}$$

where $J_{0,1}$, E_1, b_1, $J_{0,2}$, E_2, and b_2 are parameters to be specified; A_{C} is the total surface area of already-formed crystals and V_{S} is the total solution volume.

Rate of growth as a function of σ and T. Expressions for crystal growth rates have been formulated for each of the categories given in Figure 1.6 [Lewis et al., 2015]. These are formulated considering the rate-limiting step: integration or diffusion.

When integration is the rate-limiting step, rate expression are generally derived via theoretical energy paths, much as the rate of nucleation was derived in CNT. When diffusion is the rate limiting step, the rate is generally derived via Fick's law for diffusion. Without going through the derivations, simplified versions of commonly used rate expressions are presented here:

- integration-controlled growth:
  - rough growth: $G = k_g \sigma$
  - spiral growth: $G = k_g \sigma$ or $G = k_g \sigma^2$
  - birth and spread growth: $G = k_g \sigma^{5/6} \exp(-g/T^2\sigma)$
- diffusion-controlled growth: $G = k_g C^* \sigma$
- combined: $G = k_g \sigma^g$

where G is the growth rate of a given crystal (rate of change of the characteristic length per time); k_g and g are unknown constants.

A complete population balance *model*. Once the underlying mechanisms (nucleation, growth, agglomeration, etc.) have been specified along with their respective rate equations and the solubility has been characterized, a complete population balance model is obtained. An example of a population balance model for nucleation- and growth-only batch cooling crystallization is given below.

Population balance model (PBM)

$$\underbrace{\frac{\partial n(L,t)}{\partial t}+\frac{\partial\big(G(\sigma,T)n(L,t)\big)}{\partial L}=0}_{\text{1D PB, nucleation- and growth-only}} \tag{2.6a}$$

$$\underbrace{n(0,t)=\frac{J\big(n(L,t),\sigma,T\big)}{G(\sigma,T)}}_{\text{boundary condition}},\quad \underbrace{n(L,t)=0}_{\text{initial condition}} \tag{2.6b}$$

$$\underbrace{\sigma=\frac{C-C^{*}(T)}{C^{*}(T)}}_{\text{sol. and sup.}} \tag{2.6c}$$

$$\underbrace{C(t)=C(0)-\frac{1}{V}\rho k_v\int_0^{\infty}n(L,t)L^3dL}_{\text{solid-liquid mass balance}} \tag{2.6d}$$

$$\underbrace{J\big(n(L,t),\sigma,T\big)=J_{0,1}\exp(-E_1/T)\sigma^{b_1}+J_{0,2}\exp(-E_2/T)\sigma^{b_2}A_{\mathrm{C}}/V_{\mathrm{S}}}_{\text{nucleation rate model}} \tag{2.6e}$$

$$\underbrace{G(\sigma,T)=k_g\sigma^{g}}_{\text{growth rate model}} \tag{2.6f}$$

2.2.2 Reduction to moments: changing the state representation to ease the computational burden

Solving a population balance model is computationally intensive. This is because numerically solving a partial differential equation (e.g. Equation 2.6a) requires both spatial and temporal discretization. To make computation tractable, a common strategy is to reduce the state representation while maintaining the model structure from the population balance equation. The most used method for achieving this is termed the method of moments.

The method of moments: reducing the model complexity by reducing the state representation. For the simple case of nucleation- and growth-only, like that given above, a transformation can be used to reduce the state representation from the crystal size distribution density function to a series of *moments*, and thereby reduce the partial differential equation to a series of ordinary differential equations.

Moments of the size distribution are defined as follows:

$$\mu_i \equiv \int\limits_0^\infty n(L) L^i dL, \tag{2.7}$$

where μ_i is said to the the i$^{\text{th}}$ moment of the distribution $n(L)$.

Applying the moment transform to Equation (2.6a) gives us a new representation of the crystal state and lets us reduce the partial differential equation to a series of ordinary differential equations. Assuming the growth rate is size independent:

$$\int\limits_0^\infty \left(\frac{\partial n(L,t)}{\partial t}\right) L^i dL = -G \int\limits_0^\infty \left(\frac{\partial n(L,t))}{\partial L}\right) L^i dL,$$

commuting the integral and using integration by parts

$$\frac{\partial \int\limits_0^\infty n(L,t) L^i dL}{\partial t} = -G\left(L^i n(L,t)|_0^\infty - i\int\limits_0^\infty n(L,t) L^{i-1} dL\right),$$

assuming $n(\infty, t) = 0$ and applying the moment definition we arrive at

$$\frac{\partial \mu_0}{\partial t} = Gn(0,t), \text{ and}$$

$$\frac{\partial \mu_i}{\partial t} = iG\mu_{i-1} \text{ for } i \geq 1.$$

Using the boundary condition given by Equation (2.2), we get the following expressions for the rate of change of the zeroth through third moments:

Moments balance

$$\frac{\partial \mu_0}{\partial t} = J, \tag{2.8a}$$

$$\frac{\partial \mu_1}{\partial t} = G\mu_0, \tag{2.8b}$$

$$\frac{\partial \mu_2}{\partial t} = 2G\mu_1, \tag{2.8c}$$

$$\frac{\partial \mu_3}{\partial t} = 3G\mu_2. \tag{2.8d}$$

Combined with the mass balance and nucleation and growth rate expressions, this gives a new model of the crystallization dynamics.

Moments Model

$$\underbrace{\frac{\partial \mu_0}{\partial t} = J,\ \frac{\partial \mu_1}{\partial t} = G\mu_0,\ \frac{\partial \mu_2}{\partial t} = 2G\mu_1,\ \frac{\partial \mu_3}{\partial t} = 3G\mu_2}_{\text{moments balance}} \tag{2.9a}$$

$$\underbrace{\mu_0(0) = 0}_{\text{initial condition}} \tag{2.9b}$$

$$\underbrace{\sigma = \frac{C - C^*(T)}{C^*(T)}}_{\text{sol. and sup.}} \tag{2.9c}$$

$$\underbrace{C(t) = C(0) - \frac{1}{V}\rho k_v \mu_3}_{\text{solid-liquid mass balance}} \tag{2.9d}$$

$$\underbrace{J = J_{0,1}\exp(-E_1/T)\sigma^{b_1} + J_{0,2}\exp(-E_2/T)\sigma^{b_2}\mu_2}_{\text{nucleation rate model}} \tag{2.9e}$$

$$\underbrace{G = k_g \sigma^g}_{\text{growth rate model}} \tag{2.9f}$$

The mathematical model given by (2.9) is easier to solve numerically—and this is a big advantage for actually employing the model. But it is important to note that the dynamics are now expressed in terms of a different crystal state representation: the crystal state is now given by the zeroth through third moments of the size distribution. In this reduced state representation we have lost information and can no longer extract the full crystal size distribution.

Fortunately, the moments themselves have physical meaning. The zeroth moment is the number of crystals and the third moment is proportional to the total, aggregate volume of the crystals and therefore the mass of crystals. From the moments we can also obtain measures of the mean crystal size, the property we aim to control.

2.3 Using the PB Framework to Establish Size Control

The PBE as a function that connects the input temperature to the output crystal size distribution. To discuss the use of a population balance model for crystal size control, it is convenient to introduce some notation. In particular, the overall population balance model, which is actually composed of a series of equations (e.g. Equations 2.6a–f), will be denoted more succinctly by the function $\mathcal{F}_{\text{PBM}}$.

This notation is used to suggest that the population balance model can be thought of as a function. Indeed, provided we can obtain a solution to the series of equations, the population balance model does act as a function—for unseeded batch cooling crystallization, the population balance model serves a function that takes, as input, an initial solution concentration and a temperature time profile, and outputs time profiles for the crystal size distribution density function and the solution concentration:

$$\overbrace{[n(L,t), C(t)]}^{\text{output}} = \mathcal{F}_{\text{PBM}}\big(\underbrace{T(t), C(0)}_{\text{input}}\big). \tag{2.10}$$

Another was of saying and expressing this is: the population balance model predicts the time evolution of the crystal size distribution and solution concentration from a specified temperature profile and initial concentration

$$\mathcal{F}_{\text{PBM}} : T(t),\ C(0) \mapsto n(L,t),\ C(t).$$

This function is very useful for establishing control—provided that the population balance model is an accurate representation of reality and can be solved. In practice, these two qualifications are not always easy to meet. To be precise in our discussion of the use of population balance models for control, two additions to the function notation are needed.

The first is to signify the population balance model *details.* That is: the choice of underlying crystallization mechanisms deemed important (nucleation, growth, agglomeration, dissolution, etc.); the expressions chosen to represent the rate at which each of these processes occur; and, finally, the parameters in these rate expressions.

The symbol $\mathcal{M}$ is used indicate the choice of rate expressions in the PBM; the set of parameters is represented by $\theta_{\mathcal{M}}$. These are expressed in the function notation as follows:

$$[n(L,t), C(t)] = \mathcal{F}_{\text{PBM}}\big(T(t), C(0)|\underbrace{\mathcal{M}, \theta_{\mathcal{M}}}_{\text{details}}\big). \tag{2.11}$$

The second add-on needed, is to signify the computational method. As the set of equations used to express a population balance model (e.g. equations 2.6a–f) cannot be *solved* explicitly in most cases, a numerical solution is used. Sometimes this represents an approximation that should be distinguished from exact solution. The computational method and numerical solution are denoted with the following markings:

$$[\tilde{n}(L,t), \tilde{C}(t)] = \tilde{\mathcal{F}}^{\mathcal{C}}_{\text{PBM}}\big(T(t), C(0)|\mathcal{M}, \theta_{\mathcal{M}}\big), \tag{2.12}$$

where the tilde is meant to suggests an approximate solution and the symbol $\mathcal{C}$ signifies the computation method.

2.3.1 Population balance model development: choosing $\mathcal{M}$ and $\theta_{\mathcal{M}}$

Developing an accurate population balance model requires specifying which underlying crystallization mechanisms that are important and also specifying appropriate expressions for these mechanisms. So how is this done? The underlying mechanisms are selected prior to running experiments with the exception of specifying a set of unknown parameters, which are found by fitting to data. In the notation given above, $\mathcal{M}$ is chosen by the user, leaving the parameters $\theta_{\mathcal{M}}$ to be identified.

Parameter identification. To identify the parameters, $\theta_{\mathcal{M}}$, the model is fit to data. Usually this data comes in the form of a concentration profile for a batch cooling crystallization or series of batch cooling crystallizations implemented using select temperature profiles. Sometimes the concentration profile data is also supplemented with crystal size distribution data. Once the measurements are collected, an optimization program, like the following, is used to specify the unknown model parameters:

$$\underset{\theta_{\mathcal{M}}}{\text{minimize}} \quad \left\{ \sum_{k=1}^{N} \left(\hat{C}(t_k) - \tilde{C}(t_k) \right)^2 + \rho \sum_{k=1}^{N} ||\hat{n}(L, t_k) - \tilde{n}(L, t_k)|| \right\}$$

$$\text{subject to} \quad [\tilde{n}(L, t), \tilde{C}(t)] = \tilde{\mathcal{F}}^{\mathcal{C}}_{\text{PBM}}\big(\hat{T}(t), \hat{C}(0) | \mathcal{M}, \theta_{\mathcal{M}}\big),$$

where $\hat{C}(t)$ is the measured concentration profile, ρ is an adjustable parameter, $\hat{n}(L, t)$ represents the measured or, more accurately, estimated crystal size distribution, and $||\hat{n}(L, t_k) - \tilde{n}(L, t_k)||$ expresses the difference between the measured crystal size distribution and the model-predicted crystal size distribution at time t_k.

2.3.2 Population balance model-based size control

Open-loop, programmed cooling. Provided it is accurate and can be computed in a reasonable time, the population balance model function given by (2.12) is a very useful function to have in optimizing operations to produce crystals of targeted sizes.

It can, for example, be embedded in an optimization program to identify the right temperature profile for producing crystals with desired size characteristics, such as a target mean crystal size:

$$\begin{aligned} \underset{T(t),\, t\in[0,t_N]}{\text{minimize}} \quad & \left\{ \left(\bar{L}(t_N) - \bar{L}^{\oplus} \right)^2 \right\} \\ \text{subject to} \quad & \bar{L}(t_N) = \textstyle\int_0^\infty \tilde{n}(L, t_N) L \, dL, \\ & [\tilde{n}(L,t), \tilde{C}(t)] = \tilde{\mathcal{F}}^{\mathcal{C}}_{\text{PBM}}\big(T(t), C(0) | \mathcal{M}, \theta_{\mathcal{M}} \big), \\ & C(0) = C^{\text{init}}, \\ & n(L,0) = \mathbf{0}, \end{aligned} \tag{2.13}$$

where $\bar{L}^{\oplus}$ denotes the target mean crystal size.

This type of application—using a population balance model to find an optimal temperature profile—is the most frequent application of the population balance model framework for control of batch cooling crystallizations. It has been extensively investigated in the literature dating back to the early 1970's, when Mullin and Nyvlt [Mullin and Nyvlt, 1971] and then Jones and Mullin [Jones and Mullin, 1974] established that a programmed cooling profile (obtained via a population balance model) led to better CSD characteristics in comparison with linear or natural cooling. There have been numerous publications on variants of this strategy aimed at bringing the capability from *better* CSD characteristics to tightly *controlled* CSD characteristics [Rawlings et al., 1993, Miller and Rawlings, 1994, Matthews and Rawlings, 1998, Lang et al., 1999, Worlitschek and Mazzotti, 2004, Hu et al., 2005].

But the above approach faces limitations that can stand in the way of reliable control. Solving the optimization problem given by (2.13) is computationally demanding, even for the simplest population balance model. More detrimental: model-mismatch and process uncertainty severely degrade the performance of the *open-loop* control temperature profile when implemented on a real system.

Closed-loop control. The open-loop strategy may be made more robust by optimizing with explicit consideration of the model uncertainty [Nagy and Braatz, 2004, Nagy, 2009]. But this can only go so far to alleviate the shortcomings. It is more effective to close the loop and use feedback from measurements to adjust the control trajectory in real time.

Figure 2.5 illustrates a generic loop for PBM-based feedback control. The feedback loop has two key components: a *state observer*—required to infer the crystal size distribution density function from incomplete measurements—and a PBM controller, which requires an internal optimization around the population balance model function.

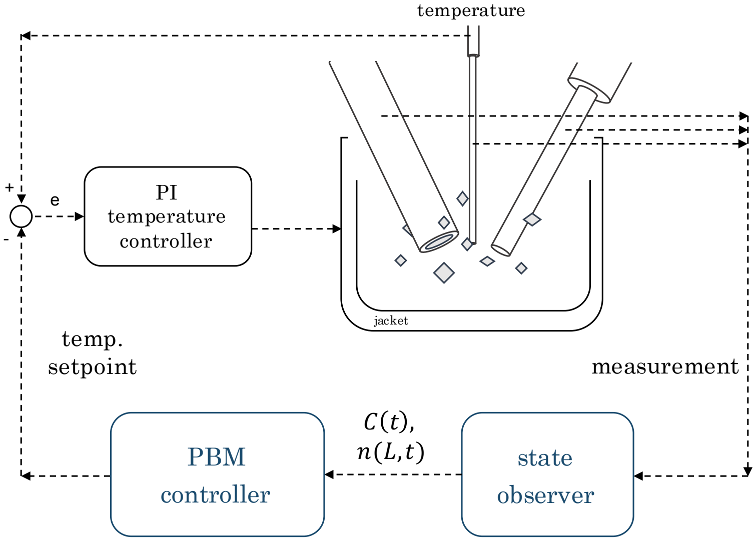

Figure 2.5: Diagram of a population balance model (PBM)-based feedback control scheme.

Although there have been numerous publications developing population balance model-based feedback control of crystallization in *simulation* as early as the 1980's [Chang and Epstein, 1987, Zhang and Rohani, 2003], experimental demonstrations of population balance model-based *feedback* control are rare [Sheikhzadeh et al., 2007, Mesbah et al., 2011, Mesbah et al., 2012] and especially sparse for batch cooling crystallization [Abbas and Romagnoli, 2006]. The lack of experimental demonstration speaks to the difficulty in applying the population balance framework to establish feedback control over the size of crystals produced by unseeded batch cooling crystallization.

2.3.3 Why I think it is difficult to use the PB framework to establish control

While the population balance framework seems a natural framework for understanding and modeling crystallization dynamics, it is difficult to apply this framework to establish control in practice. In my mind, there is a clear reason for this: the framework does not facilitate the straightforward development of feedback control and feedback control strategies are far better suited to control the endpoint of a complex dynamical system, like crystallization.

It is difficult to establish accurate open-loop control over a complex dynamical system like crystallization. Establishing accurate open-loop control over a complex dynamical system requires a very accurate model of the dynamics. That is, it requires a model that accurately predicts how the system will behave under all different inputs. Although the population balance *equation* is mathematically rigorous and elegant, the fidelity of a population balance *model* is wedded to assumptions about the underlying dynamic mechanisms (e.g. nucleation and growth). With current sensor limitations, figuring out the appropriate underlying mechanisms to include in the model and identifying rate expressions that truely capture the dynamics of these mechanisms is a challenge to say the least. As a result, it is common for population balance models to show only limited predictive ability—ultimately, undermining their use for reliable open-loop control.

The PB framework does not easily admit closed-loop control. Limitations of the population balance model predictive accuracy can potentially be overcome using feedback control. But this requires two things: online monitoring of the crystal state (as represented by the model) and quick computation. With population balance models we run into problems on both fronts. Sensors for accurately monitoring the crystal size distribution in real-time are limited. As a result, the state must be estimated by an observer, which may be of limited accuracy. Further subverting the application of population balance model-based feedback control is the complexity of a population balance model: dynamic optimization using a such a model is computationally demanding and cannot be completed in real time without significant approximation.

2.4 Chapter Conclusions

From a stirred solution of any significant volume, crystallization occurs by the formation and subsequent evolution of many crystals—that is, a *population* of crystals. The population balance provides a framework for modeling the time-evolution of a crystal population as characterized by the crystal size distribution density function, $n(L,t)$. This gives us a way of connecting the input temperature profile to the crystallization dynamics and, ultimately, the size of the produced crystals. So it seems to be an appropriate strategy for addressing the crystal size control problem.

But achieving even moderate control over the crystal size using the population balance framework requires real ingenuity. While I have seen this ingenuity out of many industrial groups and it has been published by many academic groups of the highest quality—including by former members of the research group I am a part of—, accurate crystal size control using is still not common place.

I see a reason for this: with current sensor limitations and computational capabilities, the population balance framework is not conducive to directly establishing *feedback* control, and feedback control is far better suited to address endpoint control of a complex dynamical system than open-loop control. The next chapter gives an alternative framework for addressing the crystal size control problem—one that directly enables the use of feedback.

CHAPTER 3

THE MASS-COUNT (MC) FRAMEWORK

> "All models are wrong, but some are useful."
>
> — George E. P. Box

THIS statement, made by the late statistician George Box, has come to be one of my favorite quotes. I like it because it is a reminder that you can always consider alternatives: if you understand that no model is *the truth*, then you are more free to search for a model or framework that suits the problem at hand.

In the spirit of George Box's quote, this chapter describes an alternative framework for considering crystallization dynamics and establishing crystal size control. Instead of considering crystallization dynamics in terms of the evolution of a distributed crystal population, the dynamics are expressed in terms of the evolution of just two aggregate crystal state properties: the total mass of crystals and the total number of crystals—referred to subsequently as the crystal mass and count.

The crystal mass and count serve as order parameters and give a reduced-dimensional representation of the crystal state. In recording just these two properties, we lose information and we can no longer keep track of the full crystal size *distribution*, for example. But the properties together do track the property we want to control: the mean crystal size. What is more, representing the crystal state with these two properties reveals a useful perspective that enables an intuitive understanding of the crystallization dynamics and, more to the point, feedback control.

Outline. It is worthwhile to pause here and outline how the mass-count (MC) framework will be presented and draw a parallel to the presentation of the population balance (PB) framework given in Chapter 2.

The PB framework was introduced in Chapter 2 as follows:

- the pertinent crystal state description was given (the crystal size distribution);
- followed by the perspective of crystallization dynamics that this revealed (the population balance equation); and
- finally, the chapter ended with a description of how the PB framework has been used to establish size control for batch cooling crystallization (model-based open-loop control).

Presentation of the MC framework is given a parallel organization:

- the pertinent crystal state description is given (the crystal mass and count);
- followed by the perspective of crystallization that this reveals (crystallization dynamics seen as movement in mass-count space); and
- finally, the chapter ends with a description of how the MC framework can be used to establish size control (rule-based or empirical model-based feedback control).

Following the same structure helps to link the two frameworks as different perspectives on the same process, and also highlight how these alternative perspectives induce different approaches to the crystal size control problem.

3.1 *Describing the Crystal State by the Crystal Mass and Count*

The population balance framework is predicated on representing the crystal state with the crystal size distribution density function. Representing the crystal state with a continuous distribution function is convenient because it admits a continuity equation for the rate of change of the state: the population balance equation. But we should not consider this the *only* way to represent the state—nor should we consider population balance *models* to be the truth.

Here we propose a different representation: we propose representing the crystal state with just two aggregate properties, the crystal mass and the crystal count (and also making a slight change in representing the solution state, using supersaturation in place of solute concentration). This representation is illustrated in Figure 3.1.

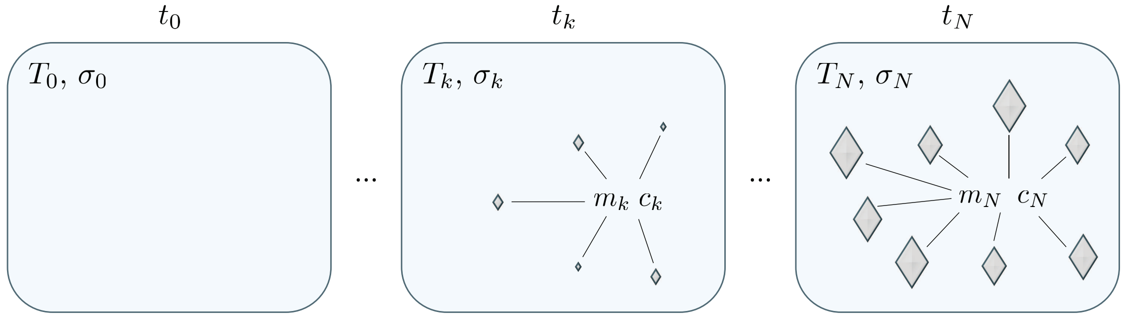

Figure 3.1: Mass-count characterization of the solution and crystal states as they evolve. At each instant in time, the solution state is indicated by the temperature, T, and the supersaturation, σ; while the crystal state is indicated by two aggregate properties: the total crystal mass, m, and the total crystal count, n.

Why this representation? The crystal mass and count do not fully capture the crystal state in the sense that these properties contain all of the information needed to fully describe the crystals. Indeed, these properties do not even capture many details of the crystal size distribution. They do, however, specify the average crystal size, and, we will find, the current crystal mass and count along with the prevailing supersaturation are enough to predict the forward change in crystal mass and count.

This means that we can write a *state-determined* dynamical system model, analogous to the population balance model but with a different state representation.

Provided these criteria are met—specification of the control property and admittance of a state-determined dynamical system model—there are clear advantages to using a low-dimensional state representation, especially a two-dimensional state representation. Using a two-dimensional state representation affords easy computation and also reveals an intuitive way to understand the dynamics: as movement in space. Aside from being low-dimensional, representing the crystal state specifically by the crystal mass and count has another big advantage for application, these properties—or properties in close relation—can be measured online. And, online monitoring, along with fast computation, facilitates *feedback* control.

Chapter 4 discusses online monitoring. The rest of this chapter reveals how the proposed mass-count crystal state representation gives us an easy way to visualize crystallization dynamics and also clarifies the route to achieving crystal size control.

3.2 Crystallization Dynamics Seen as Movement in MC Space

Let me walk through a linear cooling crystallization and describe the dynamics in terms of the evolution of the solution state properties, temperature and supersaturation, and the aggregate crystal state properties, mass and count.

The run starts with a clear (unseeded) solution at a high temperature. Cooling then creates supersaturation and, at some point, causes primary nucleation. At the onset of nucleation, both the crystal mass and count increase from zero. Continued cooling maintains supersaturation, driving crystal growth—possibly along with additional nucleation. This is seen as a further increase in the crystal mass and, possibly, the count. We can capture these dynamics by plotting each of the properties against time, as is done Figure 3.2.

Plotting the information in this way—against time—gives us a sense of how the solution state evolved and, in turn, how the crystal state evolved. But, viewed from this perspective, it is not immediately clear how to make sense of the crystallization dynamics, much less address the crystal size control problem.

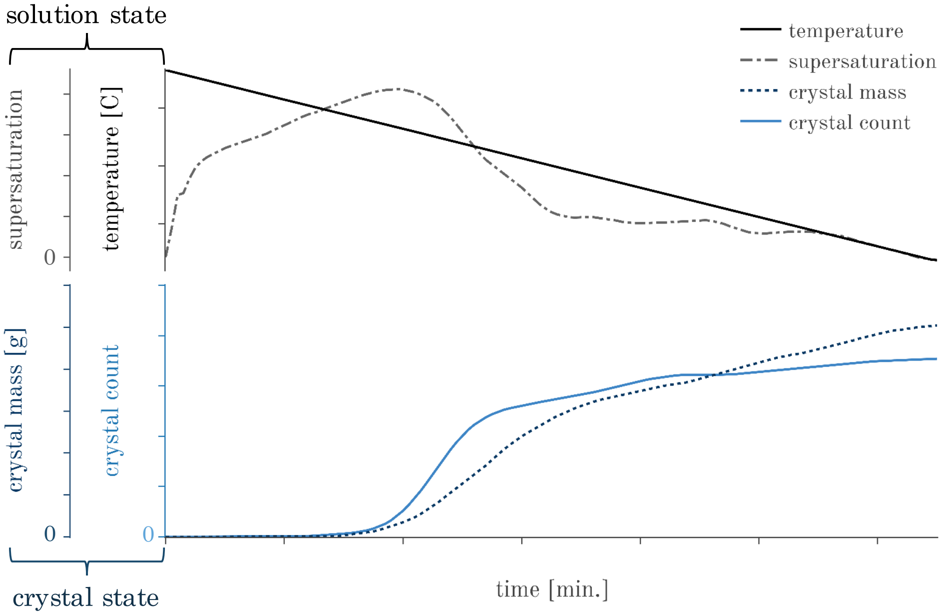

Figure 3.2: Hypothetical linear cooling crystallization as seen through time profiles of the key solution state and crystal state properties: supersaturation, temperature, crystal mass, and crystal count.

There is a better way to show the evolution of the crystal state properties—a way that reveals an intuitive understanding of the dynamics and, ultimately, helps us address the crystal size control problem. This is plotting the crystal mass against the count and viewing crystallization as a trajectory in mass-count space. Figure 3.3 shows this perspective for the same linear cooling crystallization.

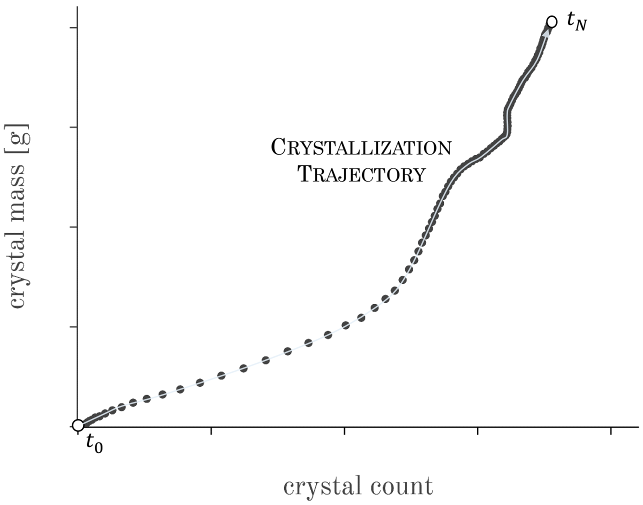

Figure 3.3: Crystallization seen as a trajectory through mass-count space (data points plotted at fixed time intervals).

Why is this perspective useful? The perspective is useful, foremost, because it shows the dynamics as movement in space. A full third of our brains may be dedicated to vision, and large portions also dedicated to space and movement. The crystallization and dissolution dynamics, when viewed from this perspective, can therefore be understood more intuitively.

Nucleation, which results in an increase in the number of crystals without a significant increase in the crystal mass, is seen as movement to the right in MC space. Growth, on the other hand, is seen as movement upwards. Conversely, dissolution is seen as movement downward and to the left. In each case, the rate of crystallization or dissolution is reflected in the rate at which the MC position changes—that is, the speed of movement. (In static images, this is reflected in the distance between data points, which are given at fixed time intervals).

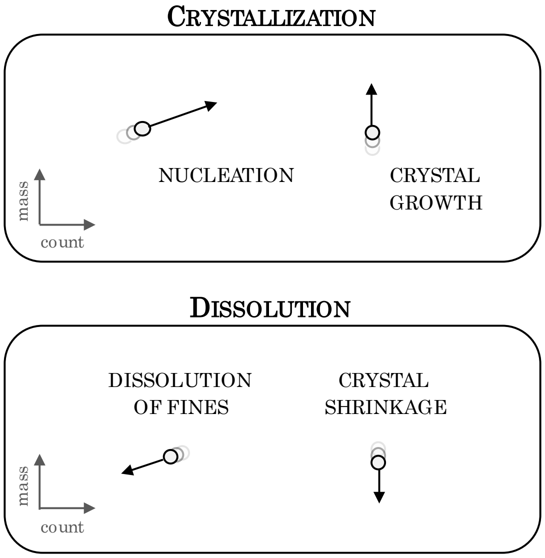

Figure 3.4: Crystallization dynamics seen as movement in MC space.

Seeing crystallization as movement in MC space can help us understand the dynamics, but this visualization—as shown in Figure 3.3 and illustrated in Figure 3.4—does not show the connection with the solution state properties that drive the dynamics. As was outlined at the end of Chapter 1, making this connection is key to addressing the crystal size control problem. To capture the connection between the solution state properties and the crystallization dynamics, we can layer the data visualization: the solution temperature and supersaturation can be indicated at each time with the color and size of the circle. This visualization is shown in Figure 3.5.

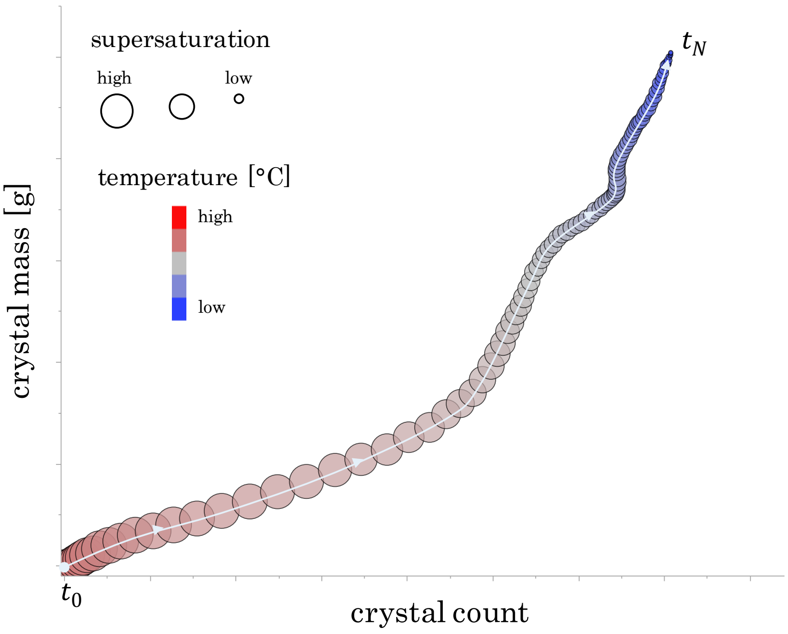

Figure 3.5: MC trajectory for cooling crystallization with the solution state variables, temperature and supersaturation, indicated by the color and size of the points (data points plotted at fixed time intervals).

Viewing the dynamics this way is not just inviting; it is also effective. In showing the crystal state evolution as a trajectory in space we are able to exploit our natural ability to understand movement; simultaneously seeing the solution state properties through color and size of the points lets us quickly connect the input action to this movement. This visualization strategy is more powerful for control than it might seem. Once we understand the crystal size control problem within the MC framework, we will see how this type of visualization can be leveraged to quickly build a mental model of the dynamics that facilitates the development of feedback control schemes.

3.2.1 Modeling movement in MC space

Although a mental model of the dynamics can take us a long way in establishing control with feedback, we cannot use a purely conceptual understanding to perform precise calculation or optimize the operation. For this, we need a mathematical model. This can be built up from first principles or constructed empirically, using data. Here an empirical or data-driven modeling strategy is suggested as it allows us to develop a mathematical model of MC movement that is better suited for the intended application: to inform feedback control.

First-principles approach to modeling the dynamics. Within the population balance framework, a mathematical model of the dynamics is built up from balance principles and knowledge of the underlying crystallization mechanisms. We might use this strategy to arrive at a mathematical model for the dynamics in terms of the mass and count. As was seen in §2.2.2, a population balance model can be reduced to a moments model, in which the crystal state is now represented by the zeroth through third moments (e.g. Equation 2.9). Since the zeroth moment is the total crystal count and the third moment is proportional to the crystal mass, this seems like a reasonable route for establishing a mathematical model of the dynamics within the MC framework.

But this modeling strategy is encumbered. As the moments model is derived from the population balance model, the fidelity of the moments model, like the population balance model it came from, is tied to the accuracy of the underlying models of nucleation, growth, dissolution, etc. This model is also troublesome for feedback control: the crystal state dimension is four in the moments model (rather than two as designed with the mass-count representation) and the solution concentration is taken to be a dependent variable. This state is more difficult to monitor in real time and, for dynamic optimization, the extra dimensions make real-time calculation more demanding.

Empirical or data-driven modeling. We can instead develop a useful model (for our control goal) by flouting first-principles-type modeling and employing an empirical or data-driven approach. To set a structure for empirical modeling, we start with the *assumption* that a state-determined, Markov model (i.e. a model that predicts the dynamics from the *current* state and input) can approximate the dynamics:

$$\Delta \mathbf{x}_\tau = F(\mathbf{x}_\tau, u_\tau)\Delta t, \tag{3.1}$$

where the state $\mathbf{x}$ is given by the crystal mass and count

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \equiv \begin{bmatrix} n \\ m \end{bmatrix}. \tag{3.2}$$

Moreover, we assume that the supersaturation, as defined by Equation 1.4, can be treated as the single *input* (manipulated by changing the solution temperature):

$$u = \sigma(T). \tag{3.3}$$

The picture associated with this model is given in Figure 3.6.

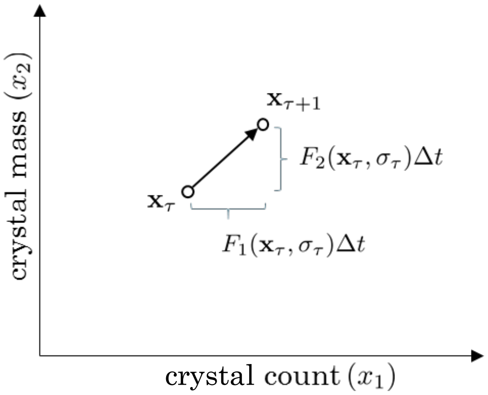

Figure 3.6: Markov model of movement in mass-count space.

Presumption of the Markov model. Even in the general form, the model given by Equation (3.1) is presumptive: it assumes that the forward movement in MC space can be predicted from the current position (mass and count) and the current supersaturation. These three properties clearly do not capture *all* details of the current state, and this therefore represents an approximation.

Treating the supersaturation as an input. A comment should be made in treating the supersaturation as an independent input, decoupled from crystal state.

The supersaturation is, of course, a function of the current concentration, which is coupled through a mass balance to the total crystal mass. However, if we can arbitrarily adjust the solution temperature and there is a one-to-one relationship between the supersaturation and the temperature, then we can also arbitrarily change the supersaturation for any given concentration and accordingly treat it as an independent variable.

Comparing the population balance model with the proposed empirical Markov model. In comparing the mathematical model given by Equation (3.1) with the population balance model given by Equations (2.6a–f), there is a stark contrast. In the population balance model, the state is infinite dimensional and the model is composed of coupled algebraic integro-partial differential equations. Moreover, the form of these equations is rigid—with the user specifying all but a set of parameters prior to running any experiments. In the proposed mass-count model, the state is two dimensional and the mathematical model consists of two ordinary differential equations written for discrete time; the form of these equations is general, with one restriction: the dynamics are a function of the current state and input.

This reduced state representation and empirical model structure does come with costs: details of the size distribution are lost and first-principles-type knowledge cannot be as easily utilized. But the model structure also pay dividends in application: the model is easy to understand, it can be learned directly from data, and optimal feedback control calculations are tractable.

3.3 Using the MC Framework to Establish Size Control

In addition to showing dynamics in an way that is intuitive to understand, the mass-count perspective clarifies the route to crystal size control. To see this, the relationship between the mass and count and the mean crystal size needs to first be made explicit.

The mass-per-count provides a measure of the mean crystal size. The total crystal mass divided by the total number of crystals—the mass-per-count—gives the mean crystal volume:

$$\bar{V}_{\text{crys}} = (m/\rho_{\text{crys}})/n,$$

where $\bar{V}_{\text{crys}}$ is the mean crystal volume, ρ_{crys} is the density of the crystal phase, m is the total crystal mass, and n is the crystal count.

This provides a natural measure of the mean crystal size, which is proportional to the cube-root of the mass-per-count:

$$\bar{L} \equiv \left(\frac{\bar{V}_{\text{crys}}}{k_{\text{S}}}\right)^{1/3} = \left(\left(\frac{m/\rho_{\text{crys}}}{k_{\text{S}}}\right)/n\right)^{1/3} \propto \big(m/n\big)^{1/3}, \tag{3.4}$$

where k_{S} is a constant shape factor.

The crystal size control problem recast as a trajectory endpoint control problem. The relationship given in Equation (3.4) indicates that the mean crystal size can be expressed as a function of the mass and count. As illustrated in Figure 3.7(a), this means that different regions in MC space correspond to crystals of different mean sizes. Moreover, the relationship suggests that a given position in mass count space corresponds with a given mean crystal size and yield, as illustrated in Figure 3.7(b).

This may not seem like a central point, but realizing that a target mean crystal size and yield corresponds a target position in mass-count space changes how we look at the crystal size control problem: the crystal size control problem is now recast as a *trajectory endpoint control problem.* And this conceptual framing of the problem clarifies the route to control.

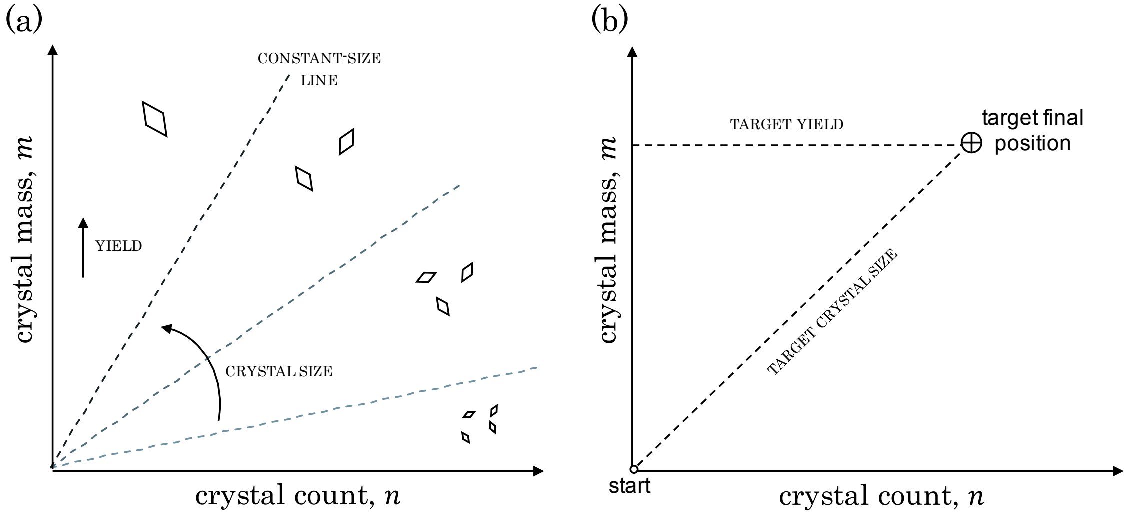

Figure 3.7: (a) Illustration showing that different regions in mass-count space correspond to crystals of different sizes. (b) Illustration showing that a given target yield and mean crystal size corresponds to a single target position in mass-count space.

Route to achieving size control using the MC framework. Once the crystal size control problem is seen as a trajectory endpoint control problem, the route to achieving crystal size control becomes clear. We need to:

1. learn to *move* in MC space; and then,
2. establish feedback control policies for reaching target positions in MC space.

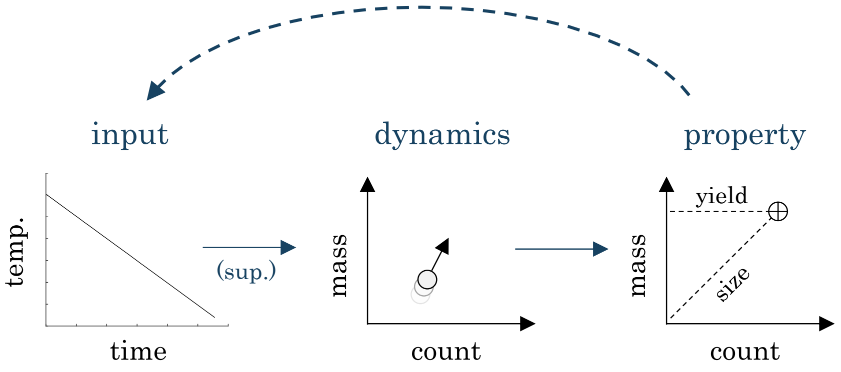

Figure 3.8: The crystal size control problem as seen through the lens of the MC framework.

3.3.1 Learning to move in MC space

A basic understanding of crystallization gives us a grasp on how to move, generally, in MC space. Cooling creates supersaturation and drives crystallization, which is seen as movement up and to the right in MC space. Conversely, heating creates undersaturation and drives dissolution, which is seen movement down and to the left in MC space.

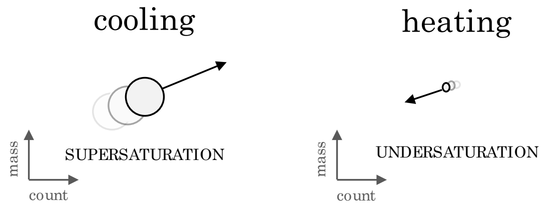

Figure 3.9: Basic understanding of the movement in MC space driven by cooling and heating.

The relationship between the speed of movement and magnitude of supersaturation (undersaturation) is also suggested by crystallization theory. High levels of supersaturation (or undersaturation) drive faster crystallization (dissolution), resulting in faster movement.

However, there are important details about MC space movement that we do not know from a basic understanding of crystallization. How, for example, do we adjust the angle of movement up and to the right during crystallization? And, what is the precise relationship between supersaturation and speed of movement? And, can more sophisticated movement be achieved through temperature cycling? Answering these questions can help set up more effective control. Fortunately, with online monitoring we can *learn* the dynamics—and answer these questions—directly from experiments.

Learning the dynamics directly from measurements is critical to the success of the MC framework; it allows us to avoid having to guess at the important underlying mechanisms. Instead, we naturally include the important dynamics: incorporating nucleation, growth, and dissolution dynamics as well as any other complex processes implicitly in the model of MC space movement we learn from experiments.

The learning process can be approached in two ways: through observation learning, in which we develop a mental model of the dynamics from visual observation; or through machine learning, in which we develop a mathematical model of the dynamics from a quantitative analysis of experimental data.

Observational learning. With real-time monitoring of the mass and count, the connection between the manipulated variable—supersaturation by way of temperature changes—and the dynamics can be intuitively learned. That is, by iteratively changing the solution conditions and watching the MC movement via visualizations like the one given in Figure 3.5, a connection can be established in much the same way that we might learn to drive. This type of intuitive learning process will be discussed in greater detail in Chapter 5, where the process will be applied to understand the dynamics of two real systems.

Machine learning. The Markov model proposed for capturing MC dynamics—i.e. $\Delta\mathbf{x}_\tau = F(\mathbf{x}_\tau, u_\tau)\Delta t$—predicts the forward change in MC position, $\Delta\mathbf{x}_\tau$, from the current MC position, $\mathbf{x}_\tau$, and degree of supersaturation, $u_\tau = \sigma_\tau$. We can apply a machine learning strategy to identify the function F, given experimental measurements of the MC movement over fixed time intervals from different starting positions under different inputs. That is, we can learn this model from training data of the following form:

$$(\mathbf{x}_1, u_1) \mapsto \Delta\mathbf{x}_1; \ldots; (\mathbf{x}_N, u_N) \mapsto \Delta\mathbf{x}_N.$$

In Chapter 5 a specific machine learning algorithm will be given and this strategy applied to two experimental systems.

3.3.2 Feedback schemes for controlling the trajectory endpoint and thereby the crystal size

A key attribute of the MC framework is that it facilitates feedback control. Two types of feedback control strategies can be developed: rule-based feedback control schemes and model-based feedback control schemes.

Rule-based feedback control. With feedback, surprisingly accurate control of a dynamical system can be achieved using only a conceptual understanding of the dynamics. There is an analogy to driving a car that brings this point home. In driving a car we are not required to maintain precise, quantitative models of the car dynamics in our head. It is enough to know the general effect of our actions: turning the wheel right alters our course in that same direction and pressing the gas peddle induces acceleration. We then constantly use feedback to adjust our actions. The MC framework enables this same type of strategy to be applied to control crystallization using only a conceptual understanding of the dynamics (obtained through observational learning).

To see this, consider the visual interface given in figure 3.10. Here the current position is shown in relation to the target position, and we can observe the most recent movement. With even the most basic understanding of movement—like, "cooling drives movement up and to the right"—this interface can facilitate feedback control. For example, we might develop a scheme in which we apply cooling whenever the current position is below and to the left of the target and heating otherwise. In Chapter 6, such rule-based feedback control algorithms are codified and tested.

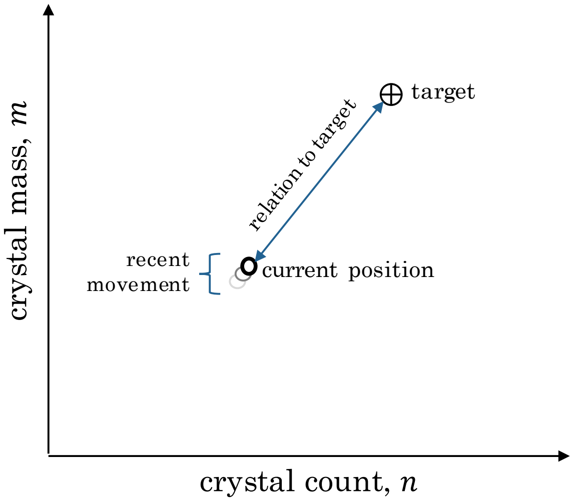

Figure 3.10: Visual interface for feedback control.

Model-based feedback control. Rule-based feedback control schemes can be powerful, but we cannot say much about the *optimality* of these schemes. To identify optimal feedback control policies, we need to identify a mathematical model of the dynamics (e.g. the proposed Markov state model) and then solve a dynamic optimization problem like the following:

$$\begin{aligned} \underset{u_\tau = \pi(\mathbf{x})}{\text{minimize}} \quad & \Phi\left(\mathbf{x}_{\tau=1,\ldots,N}, u_{\tau=1,\ldots,N}\right) \\ \text{subject to} \quad & \mathbf{x}_{\tau+1} = F\big(\mathbf{x}_\tau, u_\tau = \pi(\mathbf{x}_\tau)\big)\Delta t + \mathbf{x}_\tau,\ \tau = 0, \ldots, N-1; \\ & \mathbf{x}_0 = \mathbf{x}^{\text{init}}; \end{aligned} \tag{3.5}$$

where Φ is the objective function, F is the discrete-time dynamic model, and π represents a state-feedback policy—i.e. $\pi(\mathbf{x}_\tau) \mapsto u_\tau$. The solution to this optimization problem, π^*, is the so-called optimal *explicit model-predictive control* policy.

Here again, we find an advantage of the MC framework. With a low-dimensional model of the dynamics (F), we can apply dynamic programming to solve (3.5). The resulting optimal feedback control policy can then be applied online. In Chapter 6, this type of model-predictive feedback control will also be developed and demonstrated experimentally.

3.3.3 Limitations of the MC framework

As with the PB framework, it should be recognized that the MC framework is a perspective with limitations. Clearly, the aggregate crystal properties, mass and count, do not capture all of the details of the crystal state. While controlling the final mass and count gives us control over the mean crystal size, it does give us control over other properties of potential interest, such as the crystal size distribution width or crystal purity.

Also related to the lack of detail captured by the crystal state representation: the MC models of crystallization dynamics are somewhat removed from theories of crystallization developed at a more fundamental level. This makes it more difficult to utilize established theory to inform MC models and also more difficult to use MC models of crystallization dynamics to make inferences about the underlying mechanisms.

3.4 Chapter Conclusions

Representing the crystal state with two aggregate properties—the crystal mass and count—reveals a new framework for understanding and controlling batch cooling crystallization. Within this framework, crystallization dynamics are seen as movement in mass-count space and the crystal size control problem is re-cast as a trajectory endpoint control problem.

This conceptualization clarifies the route to establishing control: we need to (1) understand how to move in MC space and then (2) establish policies for driving to target positions. Provided online monitoring of the crystal mass and count are available, an understanding of movement in MC space can be established directly from experimental data and feedback control policies can be be developed for driving to different target positions. In Part II this strategy will be tested on two experimental systems.

A Framework for Understanding and Controlling Batch Cooling Crystallization

PART II

Application

CHAPTER 4

EXPERIMENTAL SYSTEMS AND MEASUREMENTS

4.1 Crystallization Systems Studied

Two crystallization systems are selected to demonstrate the use of the MC framework for crystal size control. The first is darapskite salt ($Na_3NO_3SO_4{\cdot}H_2O$) crystallization from a multicomponent electrolytic solution. The second is paracetamol (or acetaminophen) ($C_8H_9NO_2$) crystallization from ethanol.

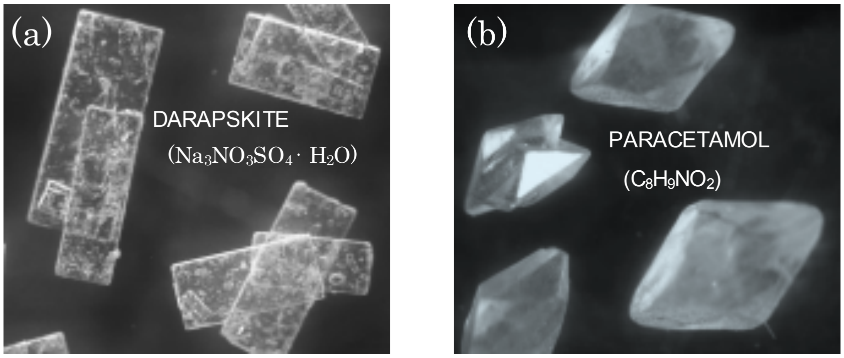

Figure 4.1: (a) Optical micrograph of darapskite salt ($Na_3NO_3SO_4{\cdot}H_2O$) crystals. (b) Optical micrograph of paracetamol ($C_8H_9NO_2$) crystals.

4.1.1 Experimental System I: darapskite salt crystallization from water

An aqueous solution containing sodium nitrate and sodium sulfate is used as a model multicomponent salt crystallization system. To be specific, 330 grams of $NaNO_3$ and 21.75 grams of Na_2SO_4 added to 300 mL of water ($\approx$ 25:1 mol-$NaNO_3$:mol-Na_2SO_4). The initial conditions for batch cooling crystallization for this system were standardized: all of our experiments started from a clear solution at 80 °C under atmospheric pressure. From this initial point, cooling caused the crystallization of a hydrated double salt, $Na_3NO_3SO_4{\cdot}H_2O$, known as darapskite (Figure 4.1(a)).

This model system came from the original motivating application for our research on crystallization control [Griffin et al., 2015d]. In particular, the relative concentrations of sodium nitrate and sodium sulfate were selected to be representative of nuclear waste compositions at the Hanford site [Nassif et al., 2008].

Background on System I selection: clean salt removal from nuclear waste. The United States Department of Energy is faced with extracting, vitrifying, and encasing the hazardous constituents contained in 56 million gallons of nuclear waste currently stored at the Hanford site in the state of Washington. Separating non-radioactive components from the waste prior to vitrification can reduce costs and expedite cleanup efforts. In particular, sodium salts—which are non-radioactive and relatively innocuous by themselves—make up a significant portion of the unprocessed waste and can potentially be removed with a crystallization-separation operation [Herting, 1996, Nassif et al., 2008].

However, the viability of such a process hinges on the ability to effectively partition the solid crystals from the solution. This, in turn, requires that the crystallization be controlled to produce large, separable salt crystals. Darapskite salt crystallization from water represents a simple, but relevant waste simulant for testing salt crystallization control strategies [Griffin et al., 2015b].

4.1.2 Experimental System II: paracetamol crystallization from ethanol

The second crystallization system selected is paracetamol crystallization from ethanol. Paracetamol, also known as acetaminophen, is a small molecule pharmaceutical that is the active ingredient in Tylenol. This system was selected for it relevance to the pharmaceutical industry.

As with system I, the initial conditions for batch cooling crystallizations or paracetamol from ethanol were standardized in this work: each run started with 200 g of ethanol containing 30 g-paracetamol/100 g-ethanol fully dissolved at 45 °C under atmospheric pressure. From this initial point, cooling caused the crystallization of paracetamol in the stable polymorphic form (Form I), which is shown in Figure 4.1(b).

A comparison of the two systems. Comparing the two systems, we see considerable differences. Darapskite is crystallized from a multicomponent electrolytic solution and forms plate-like crystals. Paracetamol is crystallized from a single-component ethanolic solution and forms monoclinic (bulky) crystals. In using these two systems as case studies, we aim to test the generality of the MC framework.

4.2 Equipment: Lab-Scale Batch Crystallizer

Batch cooling crystallizations were implemented using an OptiMax™ workstation from Mettler Toledo. This system, shown in Figure 4.2, lets us operate batch cooling crystallizations at the liter scale with accurate control over the solution temperature.

In addition, the OptiMax system provides a platform for incorporating online monitoring tools. For this work, the system was equipped with probes for focused beam reflectance measurements (FBRM) and attenuated total reflectance Fourier transform infrared (ATR-FTIR) measurements, as well as standard temperature sensors. Measurements from these instruments were recorded with iC™ software, also from Mettler Toledo, and exported in real-time to MATLAB® for processing.

4.3 Online Measurements

In the MC framework, the solution state is described by the solution temperature and supersaturation, while the crystal state is described by the aggregate properties, mass and count. Using ATR-FTIR, and FBRM, these properties—or a close relative in the case of crystal count—can be monitored in real time.

Figure 4.2: OptiMax system from Mettler Toledo equipped with probes for focused beam reflectance measurements (FBRM) and attenuated total reflectance Fourier transform infrared (ATR-FTIR) measurements.

4.3.1 Monitoring the solution supersaturation

The solution supersaturation can be monitored with ATR-FTIR measurements made in conjunction with temperature measurements [Dunuwila and Berglund, 1997].

ATR-FTIR measurements. ATR-FTIR, illustrated in Figure 4.3, measures the infrared absorbance spectrum of the solution even in the presence of suspended particles [Dunuwila et al., 1994]. As the infrared light absorbed by a solution is indicative of composition of that solution, this measurement can be calibrated to give the solute concentration.

IR-to-concentration calibration. Calibration consists of two main steps: data collection and model regression. For solutions that contain only a single solute, standard regression techniques can be applied to identify a model that correlates a single spectral feature (e.g. absorbance peak height) with the solute concentration [Dunuwila et al., 1994, Lewiner et al., 2001, Gron et al., 2002, Doki et al., 2004].

While univariate calibration models can find physical interpretations, the concentration prediction accuracy is typically improved with a multivariate calibration model

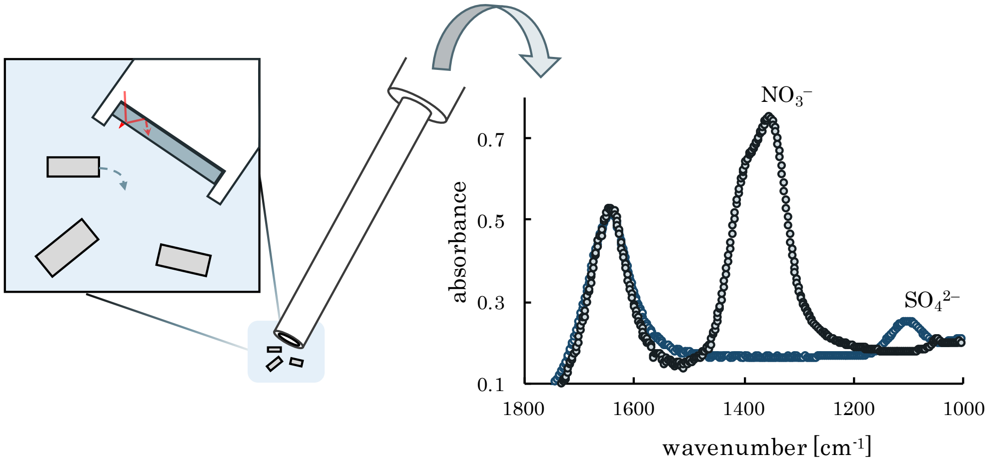

Figure 4.3: Using ATR-FTIR the *solution* infrared absorbance spectrum is measured in real time even in the presence of crystals.

that takes a number of spectral features into account [Togkalidou et al., 2001, Liotta and Sabesan, 2004, Cornel et al., 2008]. Moreover, multivariate calibration models can be used to establish concentration monitoring for multicomponent solutions [Togkalidou et al., 2002, Derdour et al., 2003].

Different types of multivariate regression algorithms are used here. To infer the concentration of both sulfate and nitrate in the multicomponent electrolytic solution, a tailored regression algorithm termed robust parameter support vector regression (RPSVR) is used [Griffin et al., 2014]. This regression algorithm, which can find broader application and is especially useful for IR-to-concentration calibration involving multicomponent solutions, is presented in detail in Appendix §A.1. To infer the concentration of paracetamol in ethanol, partial least squares regression (PLSR) is applied [Wold et al., 1984, Geladi and Kowalski, 1986].

Supersaturation from concentration and temperature measurements. For a single-component solute system, such as paracetamol in ethanol, the supersaturation, as defined in Chapter 1, is given by:

$$\sigma \equiv \frac{C - C^*(T)}{C^*(T)},$$

where C is the current solution concentration and $C^*(T)$ is the solubility concentration at the current temperature.

Given the solubility-temperature relationship, $C^*(T)$, the supersaturation, can be tracked from concentration measurements—made via ATR-FTIR—and temperature measurements.

The definition of supersaturation is more complicated for multicomponent electrolytic solutions, like that from which darapskite is crystallized. Nevertheless, the molar supersaturation [Griffin et al., 2015d]—a measure analogous to relative supersaturation—can be tracked by monitoring concentration and temperature, provided the more complicated solubility-temperature relationship is known. The molar supersaturation for darapskite is developed in Appendix §A.2.

Solubility trace. For both systems, the solubility-temperature relationship is identified using the solubility trace methodology [Barrett et al., 2010, Kee et al., 2011], which is especially useful for quickly obtaining the solubility-temperature relationship for a multicomponent solution [Griffin et al., 2015d]. The procedure is as follows: the solution is cooled to drive crystallization, allowed to equilibrate at a low temperature, and then slowly heated. During the heating stage, the solution composition, which should be approximately in equilibrium, is tracked by ATR-FTIR. The concentration-temperature curve obtained during the heating stage—like those shown for the two case systems in Figure 4.4—provides the solubility-temperature relationship.

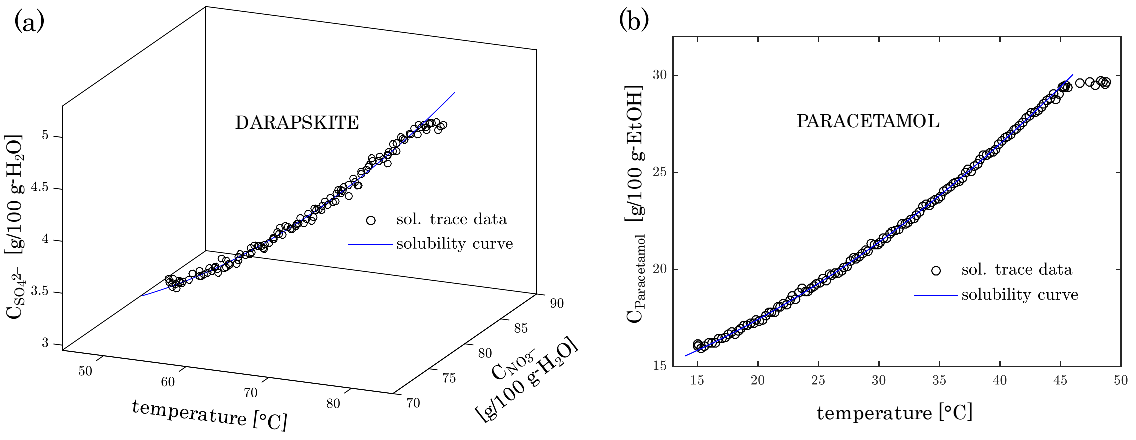

Figure 4.4: Solubility-temperature curves obtained using the solubility trace methodology: (a) darapskite salt in aqueous solution and (b) paracetamol in ethanol.

4.3.2 Monitoring the crystal mass and chord count

The crystal mass is inferred from the solution concentration. Although it is not usually done, the crystal mass can be monitored online without much additional effort provided solution concentration monitoring is already established. That is, because batch cooling crystallization is a closed system, a mass balance can be applied to infer the total crystal mass at any given time from the difference between the initial solute concentration and the current solute concentration (measured via ATR-FTIR) [Griffin et al., 2015a].

FBRM provides a measure related to the crystal count. Figure 4.5 illustrates the focused beam reflectance measurement principle: a focused light beam is scanned across the crystal-containing solution; when a crystal is in the path of the scanned beam, light is reflected from the surface and detected by the instrument. (For our particular instrument, the size limit of detection given by the vendor is 1 μm). The number of discrete reflections detected over the measurement interval (30 s in our case) is termed the chord count and denoted by the symbol c.

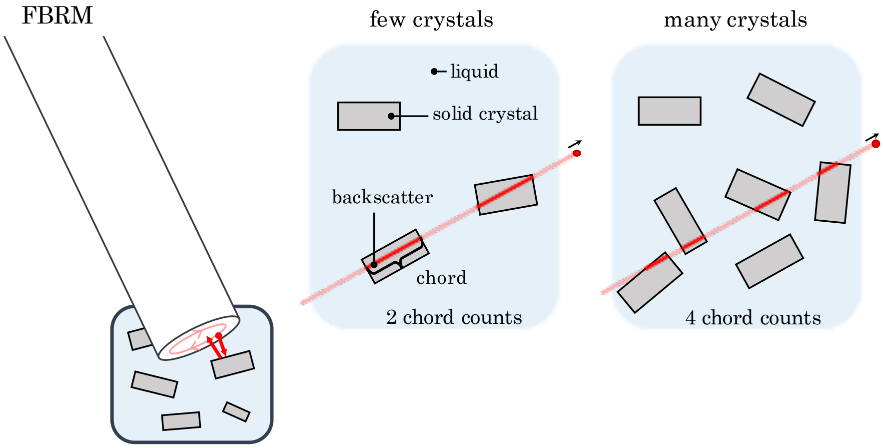

Figure 4.5: FBRM measures light backscatter from crystals suspended in solution.

The chord count recorded by FBRM is strongly correlated to the number of crystals per volume of solution—with experiments suggesting that the chord count is often approximately proportional to the number of crystals in solution for slurries with low to moderate densities [Heinrich and Ulrich, 2012, Li et al., 2013]. But the measurement is not without complications. For example, even when the chord count is seen to be proportional to the true crystal count, the proportionality constant can vary widely from system to system. This can be explained by the fact that the sensitivity of FBRM measurements to particles in solution depends on the shape of the particles as well as the optical properties of the particles. For the two studied systems we see this play out: in comparing FBRM measurements, we observe greater chord count sensitivity to paracetamol crystals than darapskite crystals—as can be explained by the fact that darapskite crystals are plate-like and more transparent.

To use the chord count measured by FBRM as a surrogate for the true crystal count, one of two relationships must be established empirically: either the chord count–crystal number relationship must be established or, more directly for our application, the mass-per-count–crystal size relationship must established.

4.4 Offline Crystal Size Measurement by Sieve Analysis

Sieve analysis is used to obtain a measure of the crystal size offline. In running this analysis, a simple two-step procedure was consistently applied: (1) the entire crystal sample was filtered from solution with a Büchner funnel and washed sparingly with cold solvent (water for darapskite and ethanol for paracetamol); (2) the crystals were then distributed evenly across the top sieve tray and the stack was placed in a RO-TAP sieve shaker for 100 min.

During shaking, the crystals are separated according to sieve diameter—with the largest crystals remaining on the top sieve tray and the smallest falling to the bottom. Weighing the mass of crystals in each sieve bin after shaking provides a measure of the crystal size distribution. An example is provided for paracetamol crystals in Figure 4.6.

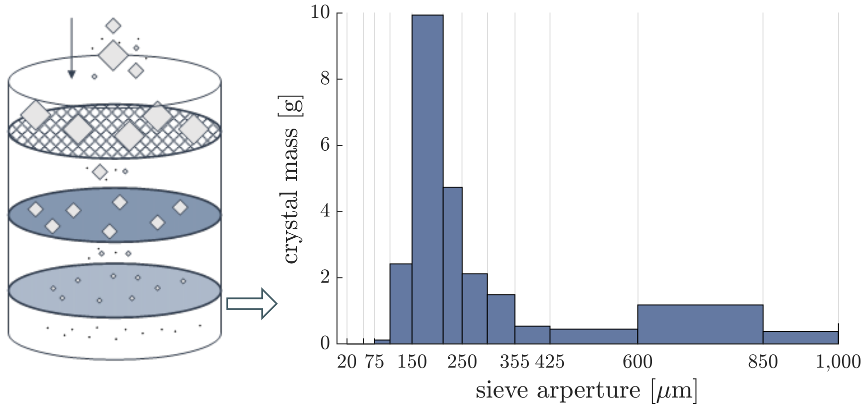

Figure 4.6: Example data from sieve analysis of paracetamol crystals.

From this data, we obtain a measure of the mass-weighted average crystal size:

$$\bar{s}_{\mathrm{MW}} \equiv \frac{\sum_{i=1}^{N_{\mathrm{bins}}} m_i s_i}{m} \tag{4.1}$$

where $\bar{s}_{\mathrm{MW}}$ is the mass-weighted average crystal size measured by sieve analysis, s_i is the midpoint of the i^{th} bin (i.e. the midpoint of the size range for the i^{th} sieve bin), m_i is the mass of crystals in the i^{th} bin, and m is the total mass of crystals.

4.5 Using the Chord Count as a Surrogate for the Crystal Count

The MC framework is predicated on the mass-count position indicating the mean crystal size. Here it is proposed that the *chord* count be used as a surrogate for the true crystal count (the total number of crystals in solution). To confirm that the chord count can be used as a surrogate for the crystal count, the relationship between the mass-per-(*chord*)count and the mean crystal size was experimentally established for the two studied systems.

4.5.1 Mass-per-count–size relationships for the experimental systems

Mass-per-count vs. mean size for darapskite crystals. Figure 4.7 shows the mean crystal size measured by sieve analysis compared with the cube root of the mass-per-count measured online for 30 different darapskite crystal samples. Despite significant scatter, analysis of variance indicates that the correlation is significant ($p < 0.0001$). This supports the use of chord count as a surrogate for the true crystal count. Moreover, the empirical relationship that is established:

$$\bar{s}_{\mathrm{MW}} = 1469(m/c)^{1/3}, \tag{4.2}$$

can be used to estimate the mass-weighted mean crystal size from online measurements of mass and count made during subsequent darapskite crystallization runs.

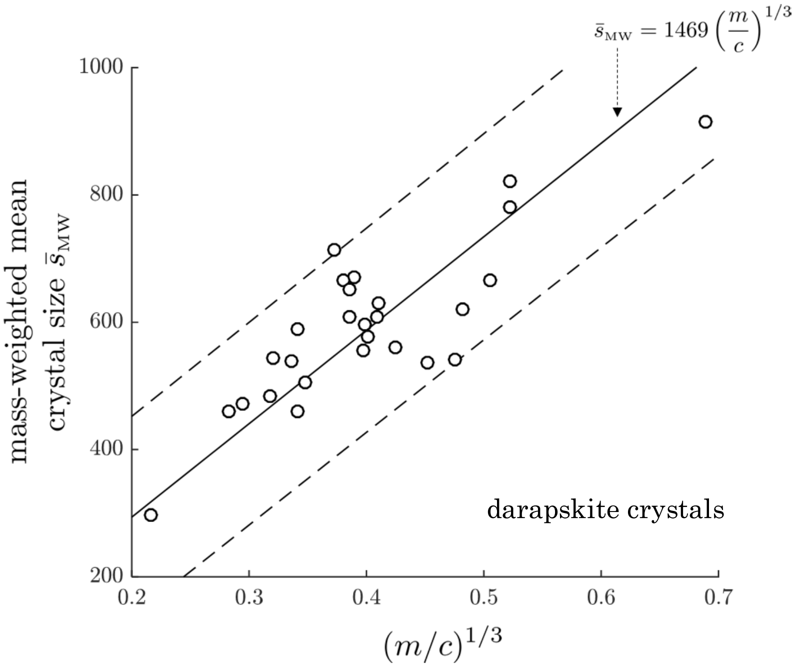

Figure 4.7: Empirical mass-per-count–size relationship observed for darapskite crystals.

Mass-per-count vs. mean size for paracetamol crystals. For paracetamol crystals, the correlation between the mean crystal size and the mass-per-count was measured for 17 different crystal samples. This correlation is shown in Figure 4.8. As with darapskite, there is some scatter in the data points. Nevertheless, analysis of variance again indicates that the correlation is significant ($p < 0.0001$)—supporting the use of chord count as a surrogate for the true crystal count. And, again, the empirical relationship identified here:

$$\bar{s}_{\mathrm{MW}} = 1305(m/c)^{1/3}, \tag{4.3}$$

can be used to estimate the mass-weighted mean crystal size from online mass and count measurements made during subsequent paracetamol crystallization runs under similar conditions.

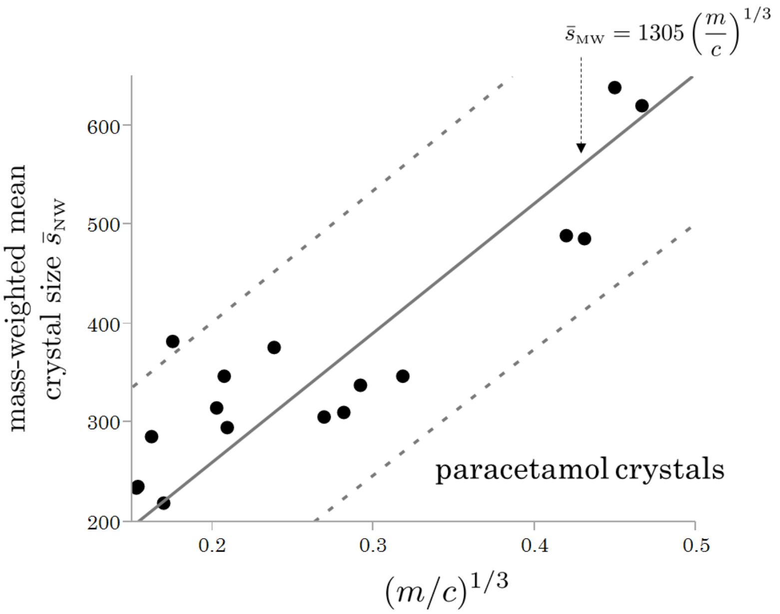

Figure 4.8: Empirical mass-per-count–size relationship observed for paracetamol crystals.

4.5.2 Mass-*chord* count space for darapskite and paracetamol

With the chord count used in place of the true crystal count, MC space is scaled. That is, the x-axis is now changed to the chord count for the particular crystallization system. As confirmed empirically by the relationships captured in Figures 4.7 and 4.8, constant-size lines still span out radially from the origin, but the exact correlation between mass and count and the size is system specific.

Figure 4.9 maps out the mass-chord count space for the two studied experimental systems. Constant-size lines are indicated with blue dashes. For darapskite salt in water, the associated mass-weighted mean crystal sizes are given by Equation (4.2); for paracetamol in ethanol, the associated mass-weighted mean crystal sizes are given by Equation (4.3).

The upper bounds of mass-count space are operation specific. The mass upper limit is the maximum yield. This can be derived from thermodynamics given the initial composition, the temperature bounds, and the solubility. For System I (darapskite salt from water), with the described initial condition and a lower temperature bound of 50 °C, a maximum yield of approximately 16 g can be achieved. For System II (paracetamol from ethanol), with the described initial condition and a lower temperature bound of 15 °C, a maximum yield of approximately 30 g can be achieved.

An exact upper limit for the chord count is less clear. The order of magnitude, however, can be established. FBRM is more sensitive to paracetamol crystals and, as a consequence, the chord count recorded for roughly the same number of crystals will be higher—this is reflected in the scale of the x-axes in Figures 4.9 (a) and (b).

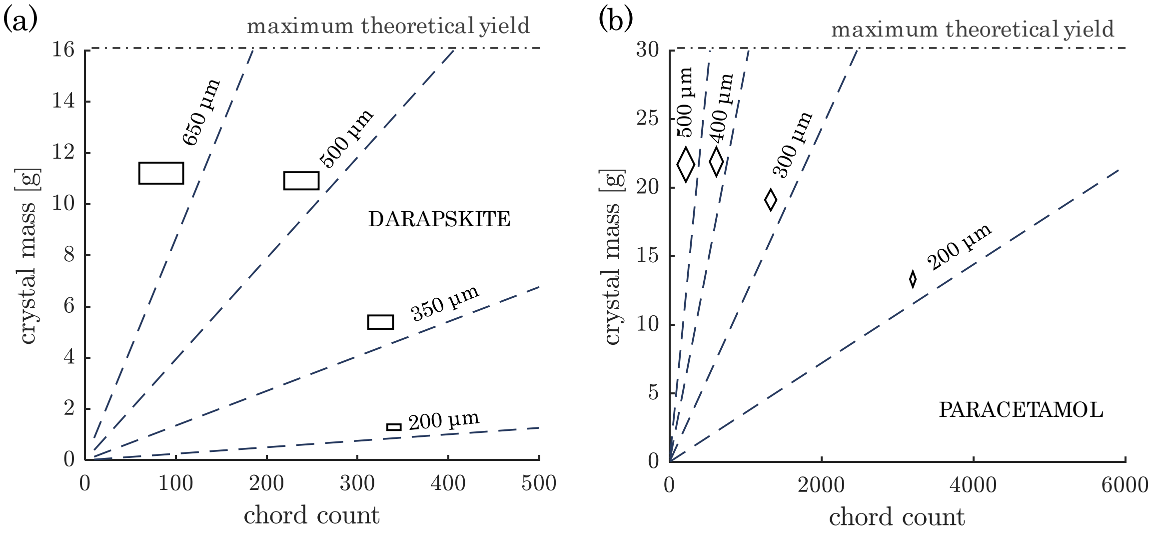

Figure 4.9: (a) Mass-chord count space mapped out for darapskite salt crystallization. (b) Mass-chord count space mapped out for paracetamol crystallization.

4.6 Chapter Conclusions

Using ATR-FTIR, FBRM and temperature measurements, the key state properties can be measured in real time. On the solution side: the supersaturation is monitored using ATR-FTIR and temperature measurements in conjunction with a calibration model and knowledge of the solubility. On the crystal side: the crystal mass can be inferred from solute concentration measurement made via ATR-FTIR and a close measure of crystal count, termed the chord count, can be measured with FBRM.

For the two experimental systems studied in this work—darapskite salt crystallization from water and paracetamol crystallization from ethanol—experimental results indicate that the mass-per-(*chord*)count measured online provides an estimate of the mean crystal size. This suggests we can apply the mass-count framework using the chord count as a surrogate for the true crystal count.

CHAPTER 5

LEARNING CRYSTALLIZATION DYNAMICS

USING the mass-count framework, it is clear what needs to be done to establish size control: we need to learn how to move in MC space and then develop control schemes for driving to target locations. This chapter is concerned with applying the first step: learning how to move in mass-count space.

Crystallization and dissolution dynamics, as seen from the MC perspective, can be learned in two ways. The first is *observational learning*—learning a set of qualitative rules from observations. The second is *machine learning*—learning a mathematical model for the dynamics from a set of training data. Both approaches are demonstrated here.

5.1 Observational Learning: General Rules for Movement in MC Space

In systems where temperature controls the supersaturation, movement in MC space is induced by changing the solution temperature. A basic understanding of crystallization and dissolution (c.f. Chapters 1 and 3) suggests some rules for this movement. Cooling builds supersaturation, driving nucleation and growth, and thereby movement up and to the right in MC space. On the other hand, heating drives dissolution, and thereby movement down and to the left in MC space. Accurate control over the final MC position, however, likely requires more detailed knowledge of the movement than: “cooling drives movement up and to the right” and “heating drives movement down and to the left.” For example, we need to understand how the rate of cooling influences the crystallization dynamics and the rate of heating influences the dissolution dynamics. We also need to understand what dynamics are induced by more complex operations that switch between cooling and heating.

We are able to learn these details, in a qualitative manner, observing the MC trajectories under varying temperature profiles.

5.1.1 Cooling operations: movement up and to the right

Linear cooling crystallization. Figure 5.1 shows the measured MC trajectory for the crystallization of darapskite salt from water driven by linear cooling. At the outset, cooling builds supersaturation, while the position remains at the origin (no crystal mass or count). Eventually the supersaturation is enough to cause primary nucleation—this is seen as movement to the right and slightly upwards in MC space. As the run progresses, the supersaturation is depleted while crystals continue to develop and grow. During this later stage, the trajectory turns upwards.

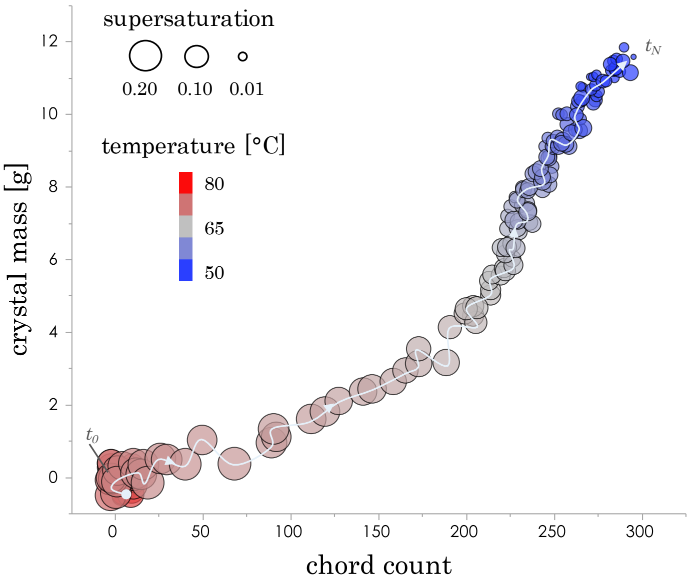

Figure 5.1: Darapskite crystallization trajectory under linear cooling at a rate of −0.25 °C/minute (positions indicated every 30 seconds).

Figure 5.2 shows a similar MC trajectory measured for paracetamol crystallization also under linear cooling. Again, supersaturation first builds and then drives nucleation and growth, which is seen as movement away from the origin.

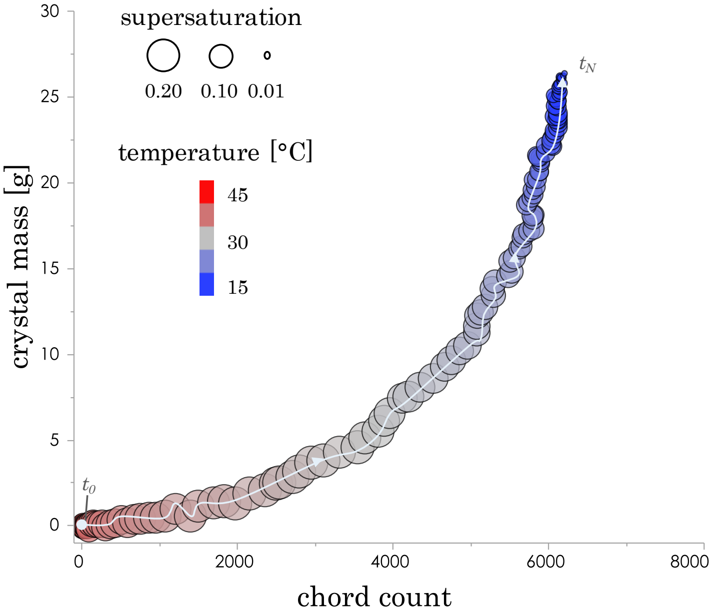

Figure 5.2: Paracetamol crystallization trajectory under linear cooling at a rate of −0.5 °C/minute (positions indicated every 30 seconds).

The two runs, shown in Figures 5.1 and 5.2, demonstrate that cooling drives movement up and to the right in MC space, as expected. This is true for all linear cooling crystallizations we have observed.

Linear cooling at different rates. The effect of cooling rate on the crystallization trajectory was probed for both systems. Figure 5.3 shows two trajectories recorded for darapskite crystallization: the first recorded for slow linear cooling (-0.25 °C/min), and the second recorded for fast linear cooling (-2.0 °C/min). Similarly, Figure 5.4 shows two trajectories recorded for paracetamol crystallization: the first under very slow linear cooling (-0.04 °C/min), and the second under under moderate linear cooling (-0.5 °C/min).

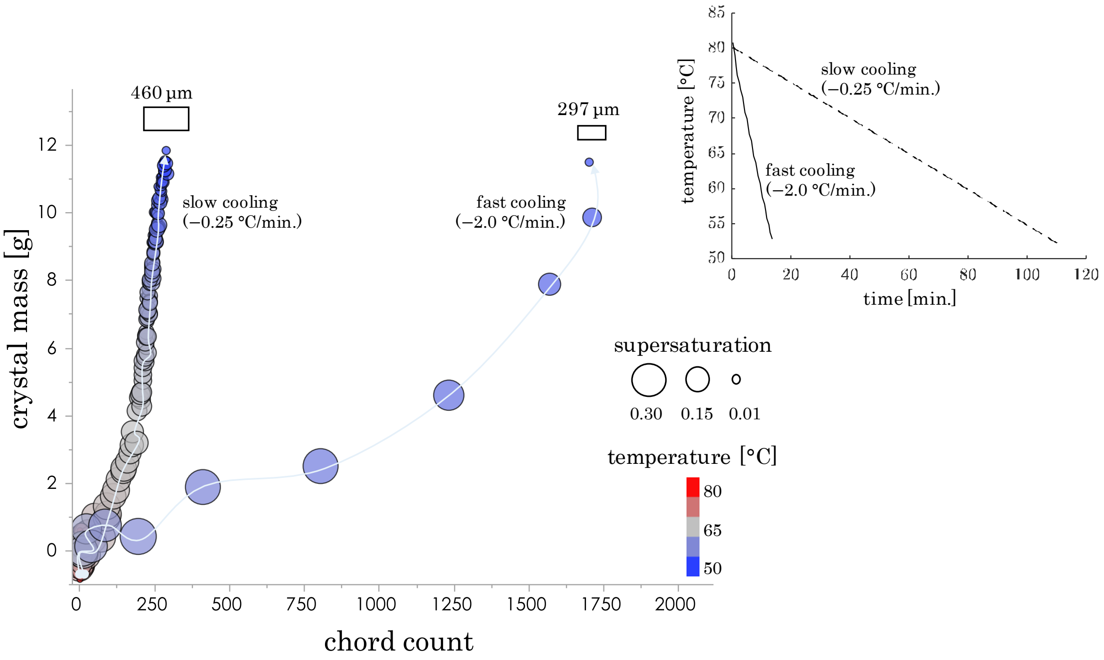

Figure 5.3: MC trajectories recorded for darapskite crystallization under slow linear cooling and under fast linear cooling (positions indicated every 30 seconds). Rectangles given at the end of the trajectory are used to represent the final mass-weighted mean crystal size determined offline by sieve analysis.

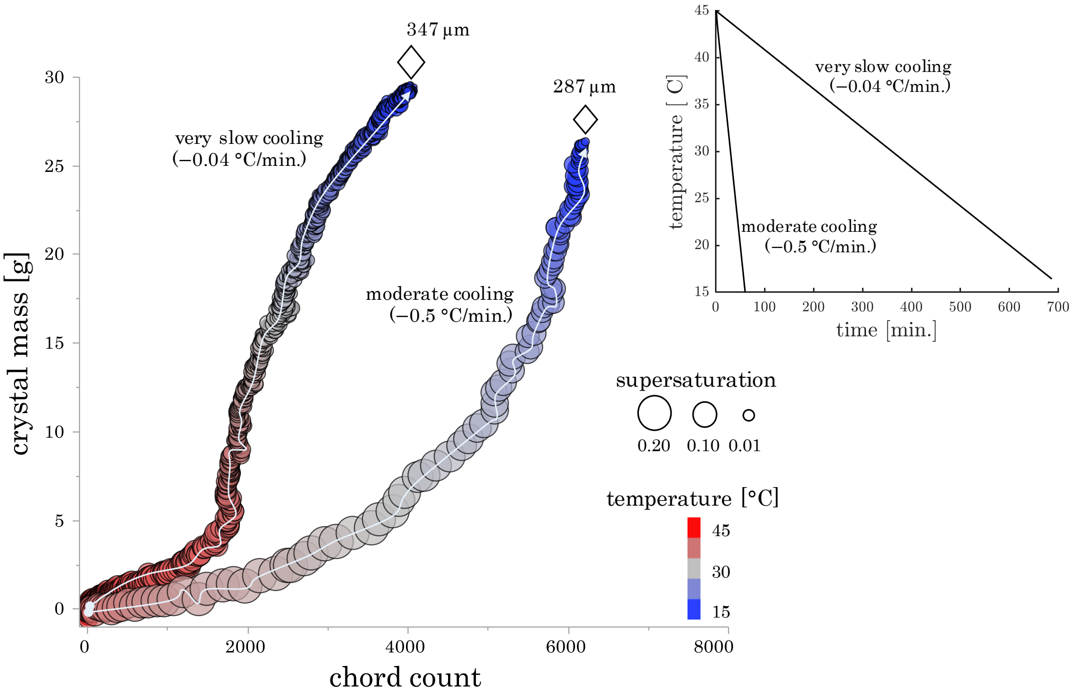

Figure 5.4: MC trajectories recorded for paracetamol crystallization under moderate linear cooling and under very slow linear cooling (positions indicated every 30 seconds). Diamonds given at the end of the trajectory are used represent the final mass-weighted mean crystal size determined offline by sieve analysis.

These figures suggest an overall effect of cooling rate on the crystallization trajectory: slower cooling results in slower movement (the points are closer together) and a *steeper* trajectory that ends at a higher mass-per-count position. The details of the trajectories, however, suggests a slight refinement to this. In each run, linear cooling is applied but the MC trajectories themselves are not linear. Instead they curve upwards as the supersaturation decreases. This indicates that it is the prevailing supersaturation, not the cooling rate itself, that dictates the speed and direction of movement in MC space, and the rule is better stated as: slow cooling results in low levels of supersaturation, which in turn results in slow movement that is directed largely upwards; conversely, fast cooling results in high levels of supersaturation, which in turn results in fast movement that is directed largely to the right.

Supersaturation control. To further confirm the effect of supersaturation on the movement in MC space, solution-state feedback control was applied to keep the supersaturation constant during select crystallization runs [Fujiwara et al., 2002]. Also, to eliminate non-linearity associated with primary nucleation, the solution was seeded.

Figure 5.5 shows the MC trajectory for seeded darapskite crystallization under a constant supersaturation. This figure indicates that constant supersaturation results in a fairly straight trajectory up and to the right, even as the cooling rate varies somewhat.

To probe the manner in which the degree of supersaturation affects the movement in MC space, multiple (unseeded) batch cooling crystallizations were operated at different levels of constant supersaturation.

Figure 5.6 shows the typical result. In this figure, crystallization trajectories are shown for two unseeded paracetamol crystallization with supersaturation maintained at 0.05 in one and 0.25 in the other. Here, it is clear that the level of supersaturation, not the cooling rate, dictates the direction of movement. Both trajectories are relatively straight after the initial nucleation event, with the trajectory produced at lower constant supersaturation being slightly steeper.

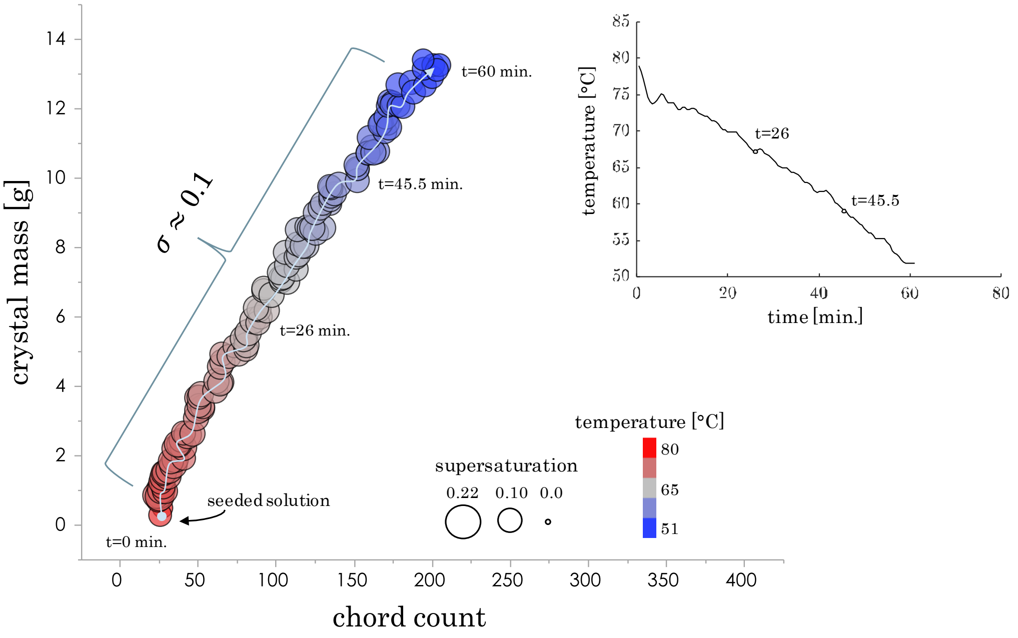

Figure 5.5: MC trajectory recorded for seeded darapskite crystallization under supersaturation control to maintain a constant supersaturation, σ, of 0.1 (positions indicated every 30 seconds).

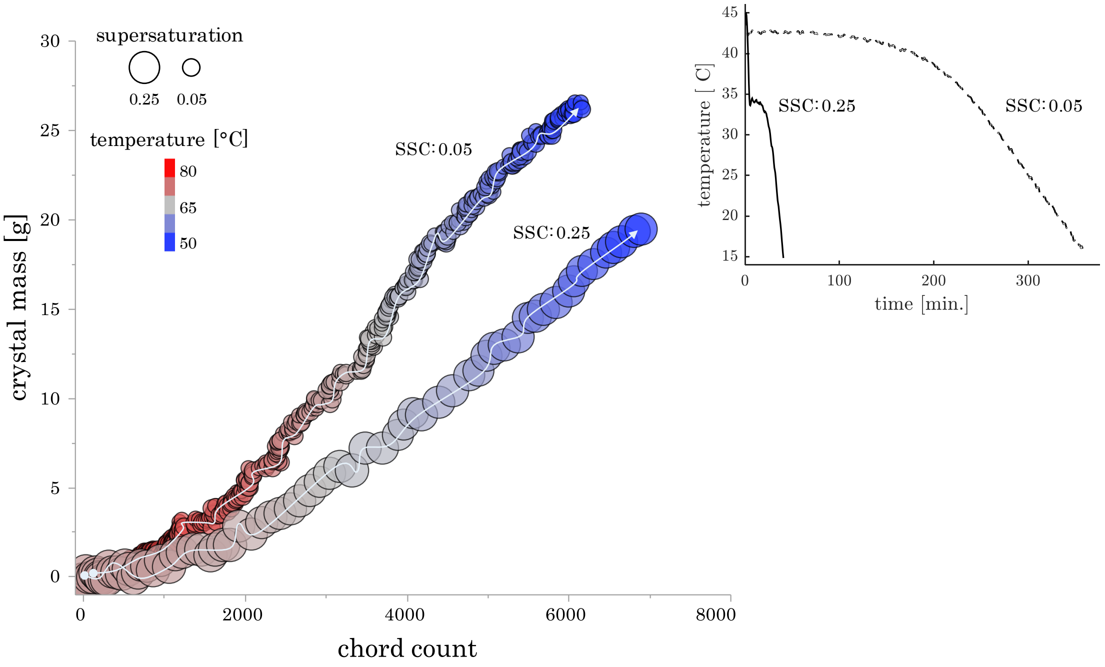

Figure 5.6: MC trajectories recorded for unseeded paracetamol crystallization with supersaturation controlled at different levels: $\sigma = 0.05$ (label SSC:0.05), and $\sigma = 0.25$ (label SSC:0.25).

Summary of MC movement during cooling-only operations. The runs shown in Figures 5.3–5.6 lead to an understanding of the MC movement induced by cooling: cooling drives movement up and to the right in MC space; the direction and speed of this movement is dictated by the prevailing level of supersaturation. Higher supersaturation drives faster movement and pushes the MC trajectory to the right, lower supersaturation results in slower movement and a steeper trajectory.

5.1.2 Heating operations: movement down and to the left

Heating creates undersaturation, causing dissolution of crystals. This is seen as movement down and to the left in MC space.

Figure 5.7 shows dissolution trajectories measured for darapskite (a) and paracetamol (b) under linear heating. In both cases, the trajectories themselves are fairly linear and movement progresses towards the origin. For crystallization, we observed that the angle of the MC movement could be influenced by the cooling rate. A key question is: can we influence the angle of movement down and to the left by adjusting the heating profile?

Here we find, unlike crystallization dynamics, that the angle of movement during dissolution is largely unaffected by the rate at which temperature is changed or the level of undersaturation. This is illustrated in Figure 5.8, which shows three dissolution trajectories for paracetamol under different heating rates. In each case, the trajectory follows the same linear path towards the origin; the heating rate only influences the rate of movement in MC space, not the direction.

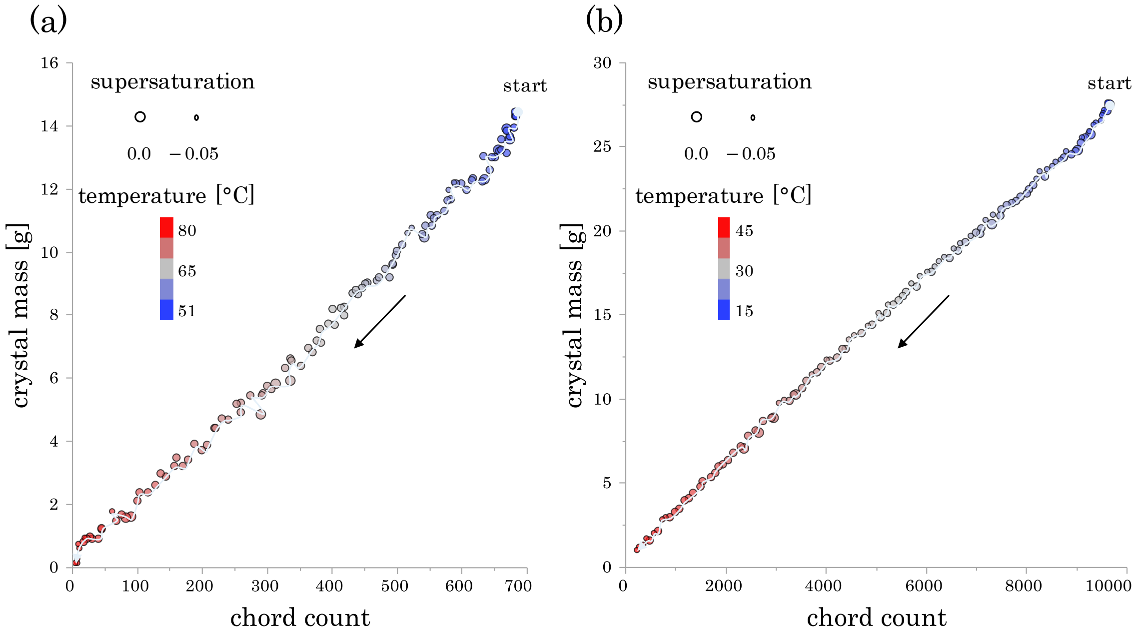

Figure 5.7: Dissolution trajectories under linear heating (0.5 °C/min.) for: (a) darapskite in water, and (b) paracetamol in ethanol. (Again, positions indicated every 30 seconds).

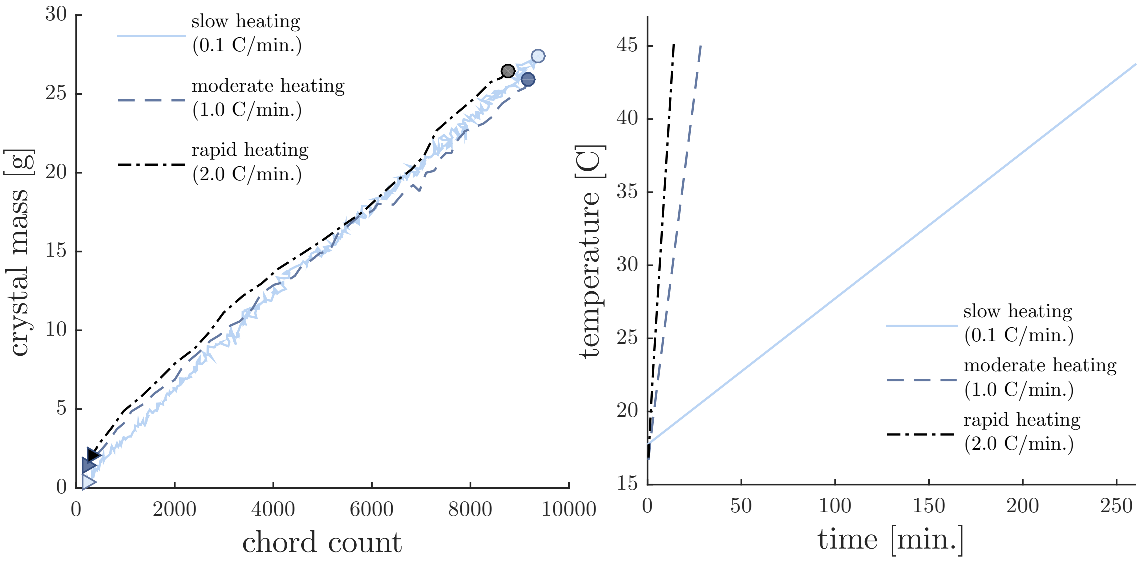

Figure 5.8: Dissolution trajectories for paracetamol under slow heating (0.1 °C/min.), moderate heating (1.0 °C/min.), and rapid heating (2.0 °C/min.).

Summary of MC movement during heating-only operations. The results in Figure 5.7 and Figure 5.8 suggest a second rule for MC movement: heating drives movement back towards the origin; this movement is faster under faster heating, but heating rate has little influence over the direction of movement.

5.1.3 Cycling between cooling and heating induces complex dynamics

The MC movement for cooling-only operations is always up and to the right. The MC movement for heating-only operations is similarly constrained, but in the opposite direction: it is always down and to the left. It turn out, however, that complex MC dynamics in different directions can be induced by combining these operations—that is, by applying cooling followed by heating and vice versa. Again, these dynamics are probed by observing the trajectories for a few select operations.

Cycling between cooling and heating ratchets the MC position upwards. Figure 5.9 shows the measured MC trajectory for darapskite when cooling was followed by rapid heating. Cooling produced high supersaturation, which initially resulted in a shallow crystallization trajectory off to the right. On heating, the supersaturation was quickly depleted and the trajectory curved upwards while the supersaturation remains positive. Further heating eventually resulted in undersaturation (at $t = 38.5$ minutes), causing dissolution and directing the trajectory back towards the origin. However, the dissolution trajectory was markedly *above* the crystallization trajectory.

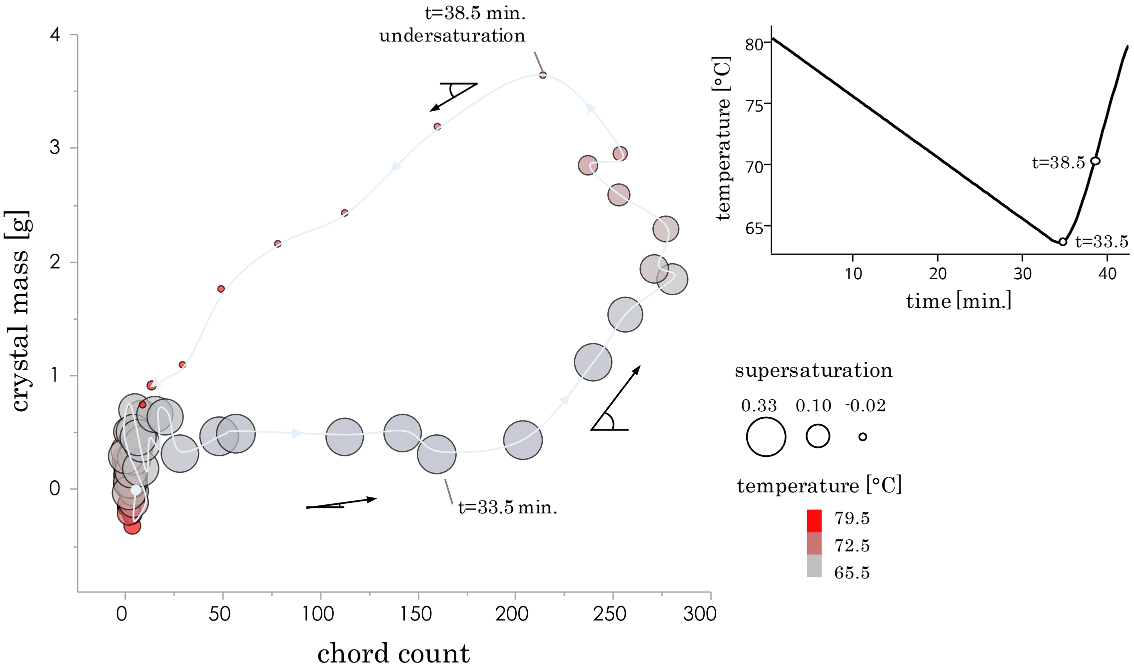

Figure 5.9: MC trajectory for the crystallization and subsequent dissolution of darapskite (positions indicated every 30 seconds).

This trajectory demonstrates how an asymmetry between crystallization and dissolution dynamics can result in net movement upwards. To further explore this, cooling-heating cycles were applied to induce crystallization and dissolution cycles while the MC trajectory was monitored. Specifically, cooling-heating cycles were implemented with feedback on the chord count following a simple rule: when the count was below a pre-picked threshold value (as it is at the start of the run), linear cooling was applied; when the count was above the threshold, linear heating was applied. Figure 5.10 shows the MC trajectory observed for darapskite and Figure 5.11 shows the MC trajectory observed for paracetamol under such operations.

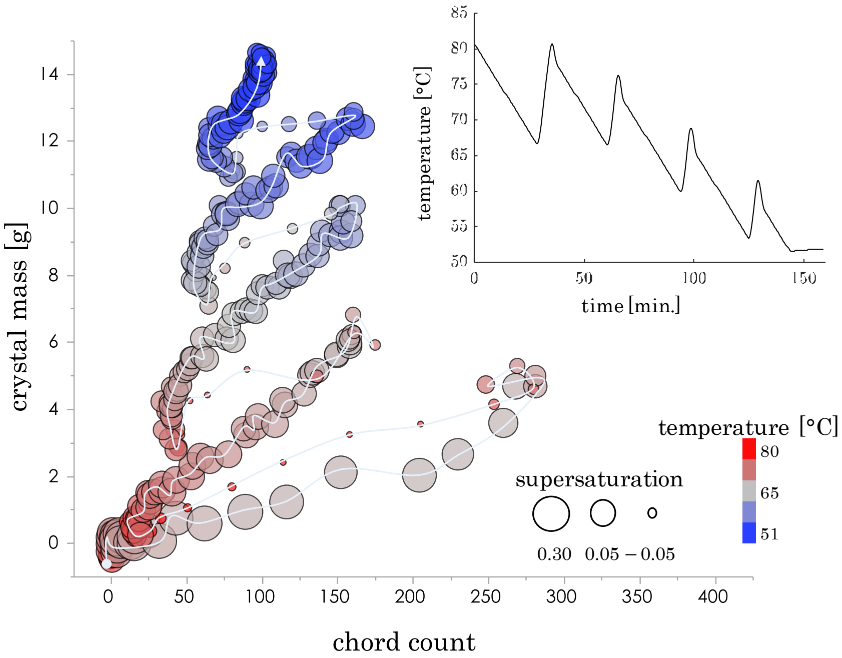

Figure 5.10: MC trajectory observed for darapskite when cooling-heating cycles were applied (positions indicated every 30 seconds).

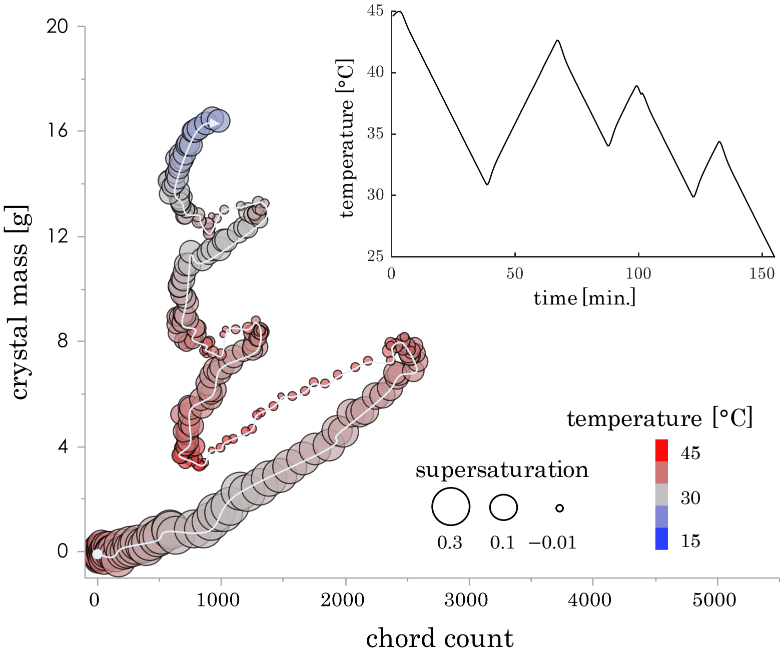

Figure 5.11: MC trajectory observed for paracetamol when cooling-heating cycles were applied (positions indicated every 30 seconds).

Summary of MC movement during cooling-heating cylces. The results given in Figures 5.9, 5.10, and 5.11 suggest one final rule for movement in MC space: cycling between cooling and heating tends to ratchet the MC position upwards.

5.1.4 General rules for movement in MC space

The general rules learned from observations for both darapskite and paracetamol are summarized in Figure 5.12. Cooling, drives movement up and to the right in MC space. The rate of cooling dictates supersaturation, which influences the speed and direction of movement: higher supersaturation results in faster movement more to the right, lower supersaturation results in slower movement more upwards. Heating, drives movement down and to the left in MC space. Here we have less actuator authority: heating, at any rate, drives movement back towards the origin; this movement is faster under faster heating, but the angle of movement is largely unaffected by heating rate. Finally, we learned that more complex trajectories can be obtained by combining cooling and heating operations. In particular, cooling-heating cycles tend to ratchet the MC position upwards—this is observed for both darapskite and paracetamol.

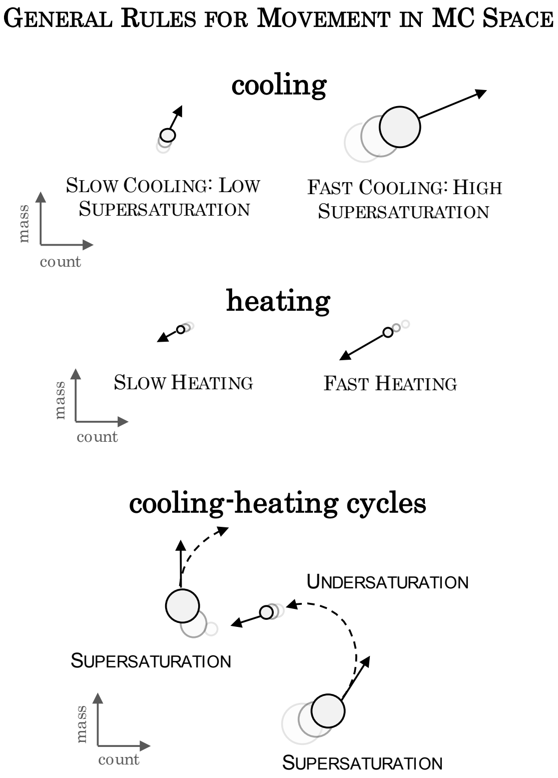

Figure 5.12: General rules for movement in MC space, observed for both darapskite from water and paracetamol from ethanol.

5.1.5 Use of qualitative MC movement rules

The rules shown in Figure 5.12 provide guidance on how to move in MC space and, combined with feedback, can be used to produce crystals of different sizes. In fact, the rules for MC movement shown here underpin the two well-known feedback control strategies for producing *larger* crystals: supersaturation control (SSC) [Fujiwara et al., 2002,Feng and Berglund, 2002,Gron et al., 2002,Liotta and Sabesan, 2004,Zhou et al., 2006,Yu et al., 2006]—in which a low level of supersaturation is maintained to promote growth over nucleation—and direct nucleation control (DNC) [Doki et al., 2004,Chew et al., 2007,Woo et al., 2009,Bakar et al., 2009b,Bakar et al., 2009a]—in which cooling-heating cycles are used to control the chord count and, ultimately, to produce large crystals.

In Chapter 6, we will see that these simple rules for movement can be combined with real-time feedback on both the mass and the count to direct the MC trajectory to target positions with a high degree of consistency, resulting in crystal size control.

5.1.6 Limitations of a qualitative MC movement rules

The qualitative understanding of dynamics developed here has limitations. It does not capture some of the more intricate details of the dynamics. For example, supersaturation is not the only factor influencing the movement; the current *crystal* state also influences the dynamics. As a clear example, the dynamics from the origin, in which no crystals exist, will be significantly different from the dynamics observed at a high mass-count position. These effects are not as easy to learn and describe with a few rules; a mathematical model is needed.

In addition, the rules provided in Figure 5.12 are for *both* of the studied systems. But the crystallization and dissolution dynamics—in terms of movement in MC space—are not *exactly* the same for these two systems. For example, the exact relationship between the prevailing supersaturation and the angle of movement is system-specific. To capture the system-specific intricacies, again a more rigorous, mathematical approach is required.

5.2 *Machine Learning: Developing a Mathematical Model of Movement in MC Space*

To capture finer details of MC movement and enable precise calculations, a machine learning strategy is applied. In principle, this approach is similar to the observational learning strategy presented: dynamic data is collected and used to learn a model. Here, however, instead of learning qualitative rules or establishing a conceptual model, a mathematical model of the dynamics is learned.

In introducing the MC framework, a Markov state model was suggested:

$$\Delta x_\tau = F(\mathbf{x}_\tau, u_\tau)\Delta t, \tag{5.1}$$

where $\mathbf{x}_\tau$ and u_τ denote state and input at the current time interval τ, and $\Delta \mathbf{x}_\tau$ represents the change in the state over the forward time interval Δt. In practice, the chord count is substituted for the crystal count

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \equiv \begin{bmatrix} \text{chord count} \\ \text{crystal mass} \end{bmatrix}, \tag{5.2}$$

and the input is taken to be supersaturation (manipulated by changing the solution temperature):

$$u = \sigma(T);$$

where σ is defined by Equation (1.4) for paracetamol and defined by Equation (A.15) for darapskite, and the solubility-temperature relationship is calculated using the solubility trace methodology (c.f. §4.3.1 and §A.3).

The object is to apply machine learning to identify the function F from collected run data for darapskite and paracetamol crystallization.

5.2.1 Machine learning

The idea of learning a mathematical model from data is straightforward in concept, if not always in practice: a training data set is first collected, then a learning algorithm is applied to identify a model that captures a relationship in the training data [Abu-Mostafa et al., 2012].

Training data. Training data, in this case, consist of chord count, crystal mass, and supersaturation measurements recorded during batch cooling crystallizations of the darapskite and paracetamol. The data is organized as a series of input-output pairs for the function F:

$$\underbrace{(\hat{\mathbf{x}}_j^\top, \hat{u}_j)}_{\text{model input}} \quad \mapsto \quad \underbrace{(\Delta\hat{\mathbf{x}}_j^\top)}_{\text{model output}}$$

where $\hat{\mathbf{x}}_j$ denotes the measured count and mass at a particular time, $\hat{u}_j$ denotes the measured supersaturation at that same time, and $\Delta\hat{\mathbf{x}}_j$ denotes subsequent change in count and mass measured over the next time interval (which is always 30 seconds in this work).

Figures 5.13(a) and 5.13(b) show the input positions in the training data sets for darapskite and paracetamol, respectively. For each of these data points, there is an associated input, $u = \sigma$, and an associated change in position, $\Delta\mathbf{x}$, recorded over the next 30 seconds. This provides the information needed to learn a Markov state model of the dynamics.

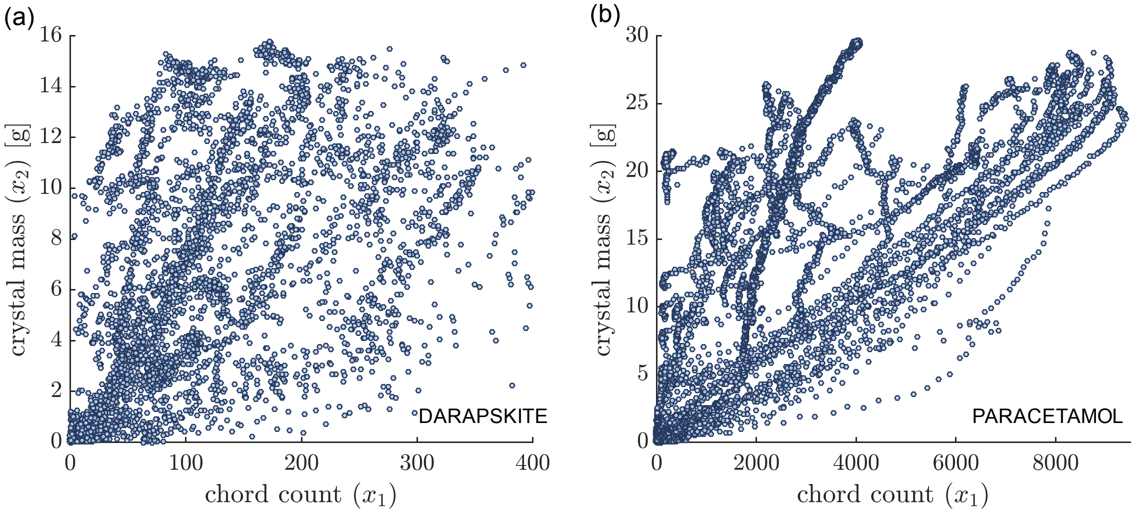

Figure 5.13: (a) Input positions in the training data set for darapskite. (b) Input positions in the training data set for paracetamol.

Learning algorithm. The learning algorithm is defined by the *hypothesis set* (i.e. the pool of candidate functions) and the *regression algorithm* (i.e. the mechanism for choosing among the candidate functions) [Abu-Mostafa et al., 2012]. Although a number of strategies exist for learning general nonlinear functions from data, our *a priori* knowledge of crystallization and dissolution motivates the use of a tailored approach [Griffin et al., 2016a]. This approach is expanded here.

Hypothesis Set

A priori we have a qualitative understanding of the crystallization dynamics that can be used to inform the function choice. We assume that supersaturation (the input u in the function F) is the primary variable—driving crystallization when positive and driving dissolution when negative. More specifically, we expect the rate of change of mass and count to:

1. be zero when the solution is saturated;
2. be positive when the solution is supersaturated and negative when the solution is undersaturated; and,

3. increase monotonically with the level of supersaturation.

This suggests constraints on the hypothesis set. That is, only functions that meet the following criteria should be considered:

1. $F(\mathbf{x}, u) = \mathbf{0}$ if $u = 0$ for any $\mathbf{x}$;

2. $uF(\mathbf{x}, u) \succeq \mathbf{0}$ for any $\mathbf{x}$; and,

3. $\partial F/\partial u \succeq \mathbf{0}$ for any any u at a fixed $\mathbf{x}$.

These constraints narrow down the possible functions somewhat, but still leave an uncountable number of different functions to choose from. To make the computation possible, we have to choose a set of generic functions. For our application, a 6^{th}-order polynomial function of u provides enough flexibility and easy computation. Taking this generic function class and adding the listed constraints yields the following hypothesis set for F:

$$\mathcal{H} = \Big\{F \big| F(\mathbf{x}, u) = [u\ u^2\ \ldots\ u^6]\beta(\mathbf{x}),\ \beta(\mathbf{x}) \in \mathbf{R}^{2\times 6},\ uF(\mathbf{x}, u) \succeq \mathbf{0},\ \text{and}\ \partial F/\partial u \succeq \mathbf{0}\Big\}.$$

Regression Algorithm

Once a hypothesis set of candidate functions has been selected, a regression algorithm must be specified to pick the function that best represents past data. We use a locally-weighted, least-squares regression algorithm. This algorithm can be expressed as follows:

for a given position, $\acute{\mathbf{x}}$

$$\begin{array}{ll} \underset{F}{\text{minimize}} & \left(\sum_{j=1}^{N_{\text{train}}} w(\acute{\mathbf{x}}, \hat{\mathbf{x}}_j; \kappa) || F(\acute{\mathbf{x}}, \hat{u}_j)\Delta t - \Delta\hat{\mathbf{x}}_j ||_2^2 \right) \\ \text{subject to} & \mathbf{F}(\acute{\mathbf{x}}, u) \in \mathcal{H}, \end{array}$$

where N_{train} is the number of samples in the training data set, $w(\acute{\mathbf{x}}, \hat{\mathbf{x}}_j; \kappa)$ is a weight function with one adjustable parameter denoted by κ, $\hat{u}$ represents the measured supersaturation, and $\Delta\hat{\mathbf{x}}$ represents the measured change in state over the corresponding forward time interval.

Overall Learning Algorithm

Combining the hypothesis set and the regression algorithm, the overall learning algorithm can be written explicitly as follows:

for a given position, $\acute{\mathbf{x}}$

$$\beta^{[\acute{\mathbf{x}}]} = \underset{\beta \in \mathbf{R}^{2\times 6}}{\operatorname{argmin}} \left(\sum_{j=1}^{N_{\text{train}}} w(\acute{\mathbf{x}}, \hat{\mathbf{x}}_j; \kappa) || [\hat{u}_j, \hat{u}_j^2, \ldots, \hat{u}_j^6] \beta \Delta t - \Delta\hat{\mathbf{x}}_j ||_2^2 \right)$$

$$\begin{aligned} \text{subject to} \quad & u[u \; u^2 \ldots \; u^6]\beta \succeq \mathbf{0}, \;\; \text{for all } u; \\ & [2u \; 3u^2 \; 4u^3 \; 5u^4 \; 6u^5] \begin{bmatrix} \beta_{2,1} & \beta_{2,2} \\ \beta_{3,1} & \beta_{3,2} \\ \beta_{4,1} & \beta_{4,2} \\ \beta_{5,1} & \beta_{5,2} \\ \beta_{6,1} & \beta_{6,2} \end{bmatrix} \succeq \mathbf{0}, \;\; \text{for all } u. \end{aligned} \tag{5.3}$$

Additional details on the training data, weighted least-squares regression, and the solution method for (5.3) are given in Appendix C.1.

5.2.2 Resulting mathematical models

The learning algorithm given by (5.3) identifies local model coefficients, denoted $\beta^{[\acute{\mathbf{x}}]}$. These coefficients specify the local dynamic model around the position $\acute{\mathbf{x}}$:

$$\left.\frac{\Delta \mathbf{x}}{\Delta t}\right|_{\acute{\mathbf{x}}} = \mathbf{F}(\acute{\mathbf{x}}, u) = \beta_1^{[\acute{\mathbf{x}}]} u + \beta_2^{[\acute{\mathbf{x}}]} u^2 + \cdots + \beta_6^{[\acute{\mathbf{x}}]} u^6.$$

For different positions, these coefficient change. As a result, a global model ($\mathbf{F}$ for all $\acute{\mathbf{x}}$) cannot be written in a compact form. This model can, however, be expressed visually.

MC Markov state model identified for darapskite. Figure 5.14 provides a visualization of the model of dynamics learned for darapskite salt crystallization and dissolution in water. The arrows depict the model-predicted movement over a 30 second time interval at different levels of undersaturation and supersaturation.

As expected, the model predicts movement up and to the right when the solution is supersaturated and predicts movement down and to the left when the solution is undersaturated. It also indicates that the angle up and to the right becomes more shallow (to the right) under higher supersaturation and the movement becomes faster (the arrow is longer). When the solution is undersaturated, however, the angle of movement is not noticeably affected by the level of undersaturation; instead, the angle seems to be affected by the position (the crystal state) while the level of undersaturation affects the speed of movement, seen by the length of arrow.

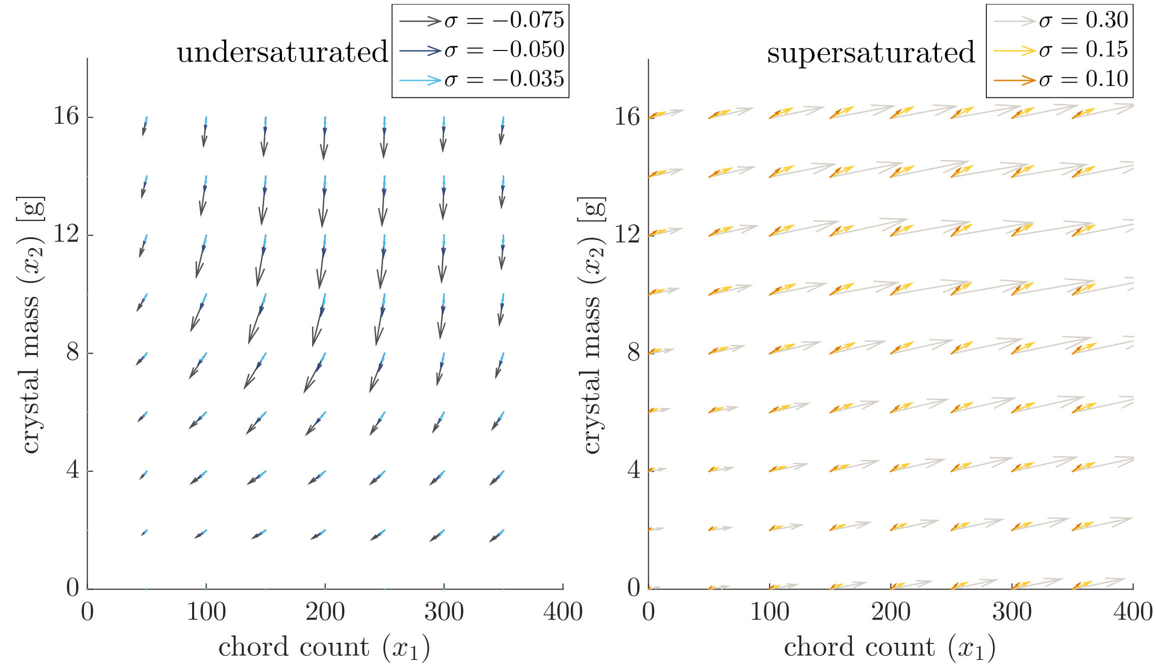

Figure 5.14: Visual representation of the MC Markov state model for darapskite salt crystallization and dissolution. Arrows depict the model-predicted movement over a 30 second time interval when the solution is (*left*) undersaturated or (*right*) supersaturated.

MC Markov state model identified for paracetamol. Figure 5.15 uses the same visualization to display the model learned for paracetamol crystallization and dissolution in ethanol. Again, the model suggests movement up and to the right when the supersaturation is positive and predicts movement down and to the left when the solution is undersaturated. Here, however, we note a difference: the model for paracetamol indicates a smaller effect of the level of supersaturation on the angle of movement during crystallization. (The arrows all point in roughly the same direction except for at very low levels of supersaturation). This suggests that we have less influence over the MC space movement for the paracetamol system than we do for the darapskite system.

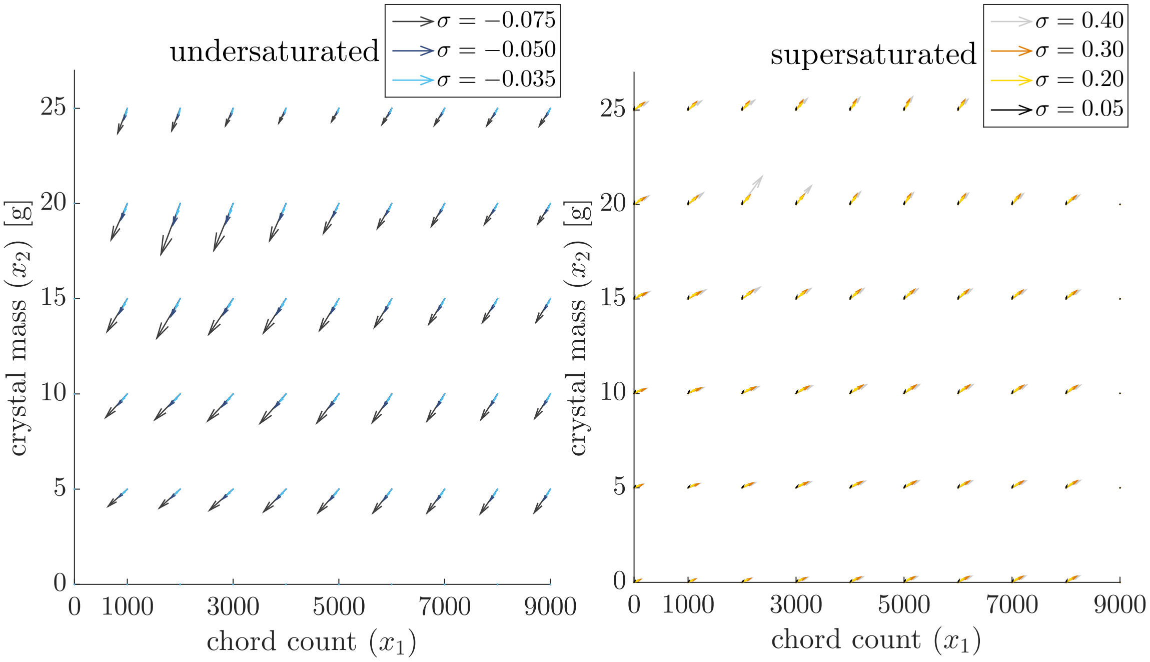

Figure 5.15: Visual representation of the MC Markov state model for paracetamol crystallization and dissolution. Arrows depict the model-predicted movement over a 30 second time interval when the solution is (*left*) undersaturated or (*right*) supersaturated.

5.2.3 Using the Markov model of dynamics to inform run design

The empirical models developed within the MC framework are intended to enable optimal, model-predictive feedback control over the final mass-count position. Before being applied for feedback control, however, they can also be used to calculate optimal open-loop policies, evaluate reachability questions, and run stochastic simulations. Results from these analyses can provide valuable insight for crystallization run design. The main takeaways are reported here, with details given in Appendix §C.2.

Calculations with the MC model for darapskite salt crystallization reveal three important points for run design:

1. temperature cycles are *required* to reach certain mass-count positions and produce crystals with a mean size greater than 750μm (Figure C.2);

2. there is a near linear tradeoff between the minimum batch time and the target

crystal size (Figure C.6); and

3. feedback control can be used drive the system towards the target, even when there is substantial measurement error, actuation delay, and a purely random component in the dynamics (§C.2.3).

Similar calculations with the model for paracetamol crystallization reveals similar conclusions, with two notes: temperature cycles are more crucial—that is, the use of temperature cycles exapands the reachable region for paracetamol to a greater extent—and producing paracetamol crystals of the same mean size takes longer.

5.2.4 Limitations of the MC model

Again, the mathematical models constructed here are built for a specific purpose: to inform model-predictive feedback control over the final MC position (and thereby control the mean crystal size). The adequacy of the models should be judged by their utility for this purpose—and will be in Chapter 6—but it is also worthwhile to point out some of the general limitations of the model framework and evaluate how well the developed models align with experimental observations for the two systems studied.

The general Markov state model of dynamics, given by Equation (5.1), predicts the forward change in the crystal mass and count from only the *current* mass, count and supersaturation. Moreover, the function is constrained—by our choice of hypothesis set—to predict an increase in *both* the mass and count when the solution is supersaturated and a decrease in *both* the mass and count when the solution is undersaturated. This is an approximation of reality.

In reality, crystal properties not captured by the mass and count may influence the dynamics, and the system history may also have an effect. In addition, although thermodynamics suggests the crystal mass will increase when the solution is supersaturated and decrease when the solution is undersaturated, it is less clear that the change in chord count must follow these same constraints.

We therefore expect the models predictions to deviate somewhat from the experimentally observed dynamics. This is true for both models, but we find that the model-experiment deviations are more evident for the paracetamol system.

5.2.5 Complex paracetamol dynamics: model-experiment mismatch

There are notable cases in which the observed dynamics for paracetamol deviate significantly from the model. These deviations are associated with complex dynamics that cannot be completely captured given the constraints placed on the model. In particular, the MC dynamics for paracetamol are complex around cooling and heating switch points at high-mass, high-count positions. Examples of these dynamics are illustrated here.

Increase in count on heating. Fig. 5.16(a) shows a select portion of a trajectory, in which a slurry was heated from 15 °C (after previously being cooled to from 45 °C to cause crystallization and generate the slurry). During heating, unexpected MC movement was observed: as the supersaturation decayed towards zero (at $t = 12$ minutes) and eventually the solution became undersaturated, the trajectory turned to the right and the chord count continued to increase.

This observation is difficult to understand from first principles, but it is consistent; that is, in every run that followed that same protocol, a jump in the chord count was observed at the onset of undersaturation. The mathematical model proposed cannot capture this phenomenon.

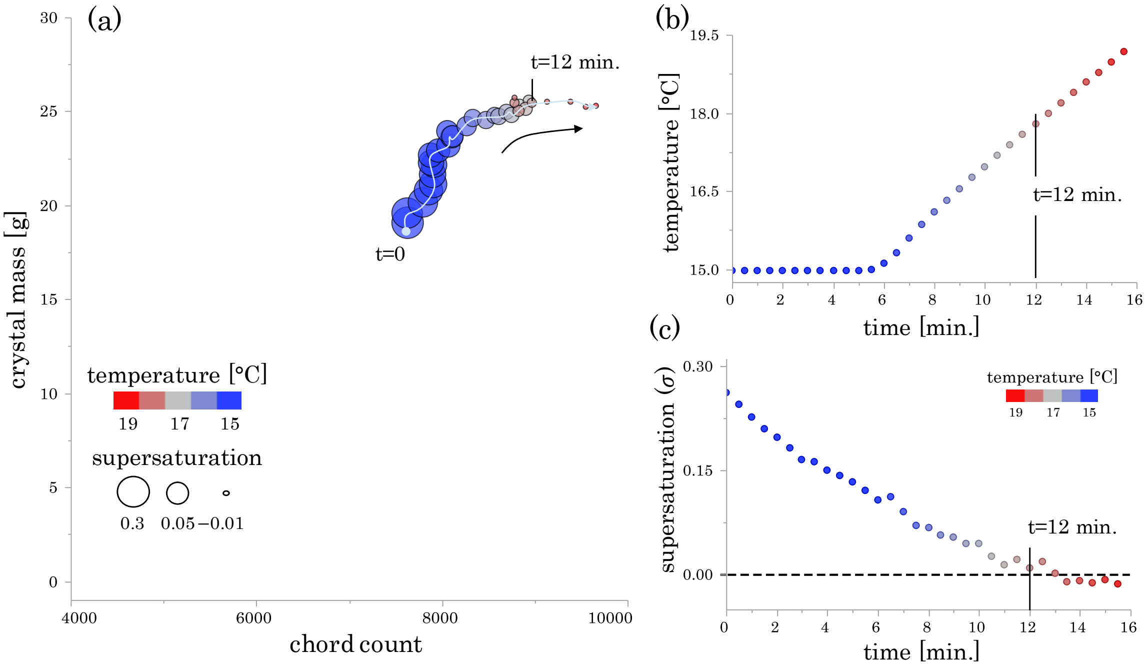

Figure 5.16: *Increase in count on heating.* (a) MC trajectory resulting from heating a slurry, with the solution initially supersaturated; (b) the temperature profile; (c) the supersaturation profile.

Ostwald ripening. The increase in chord count on heating in not the only example of complex MC dynamics observed for paracetamol around cooling-heating switches. Fig. 5.17(a) shows the trajectory induced by heating and then cooling from a high-mass, high-count position in which the solution is initially undersaturated.

Here, the initial heating, from $t = 0$ to $t = 20$ minutes, maintained undersaturation and caused dissolution. From Figure we see that the initial heating resulted in movement down and to the left, as expected. After this heating stage, cooling was applied creating supersaturation (at $t = 23$ minutes); as the solution became supersaturated the mass increased as expected but, counter-intuitively, the chord count continued to decrease from 23 to 100 minutes. This observation was also consistent from run to run. And, in this case, we can hypothesize a coherent explanation for the complex MC movement: in switching to cooling after heating, small crystals continue to dissolve and the solute redeposits onto growing larger crystals—a phenomenon known as Ostwald ripening.

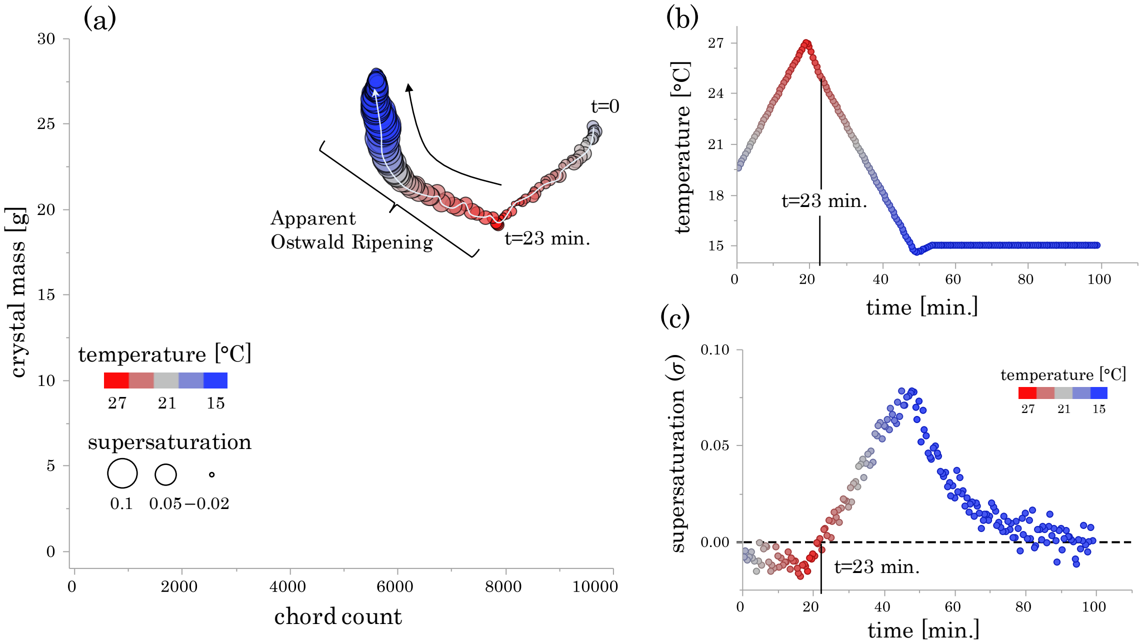

Figure 5.17: *Ostwald ripening.* (a) MC trajectory resulting from the application of a heating-cooling cycle starting from a high-mass, high-count position, with the solution initially undersaturated; (b) the temperature profile; (c) the supersaturation profile.

Dissolution of fines. A final example of complex paracetamol dynamics is demonstrated by the run shown in Figure 5.18. This run started from a high mass-per-count position—presumably indicating large crystals and no fines. Cooling from this position induced significant secondary nucleation, moving the MC position up and significantly to the right. From this point (at $t = 87$ minutes), heating then resulted in very rapid movement to the left (as seen by the large space between points).

There is a logical explanation for this. At the high-mass, high-count position reached at 87 minutes, there was a bi-modal crystal size distribution—some large crystals that made up most of the mass and many fines that made up most of the counts. Heating causes the fines to dissolve, drastically and rapidly reducing the count without significantly reducing the crystal mass. In this case, although *we* might have intuitively predicted such movement based on the history of the run, the *model* cannot as it only takes into account the current state.

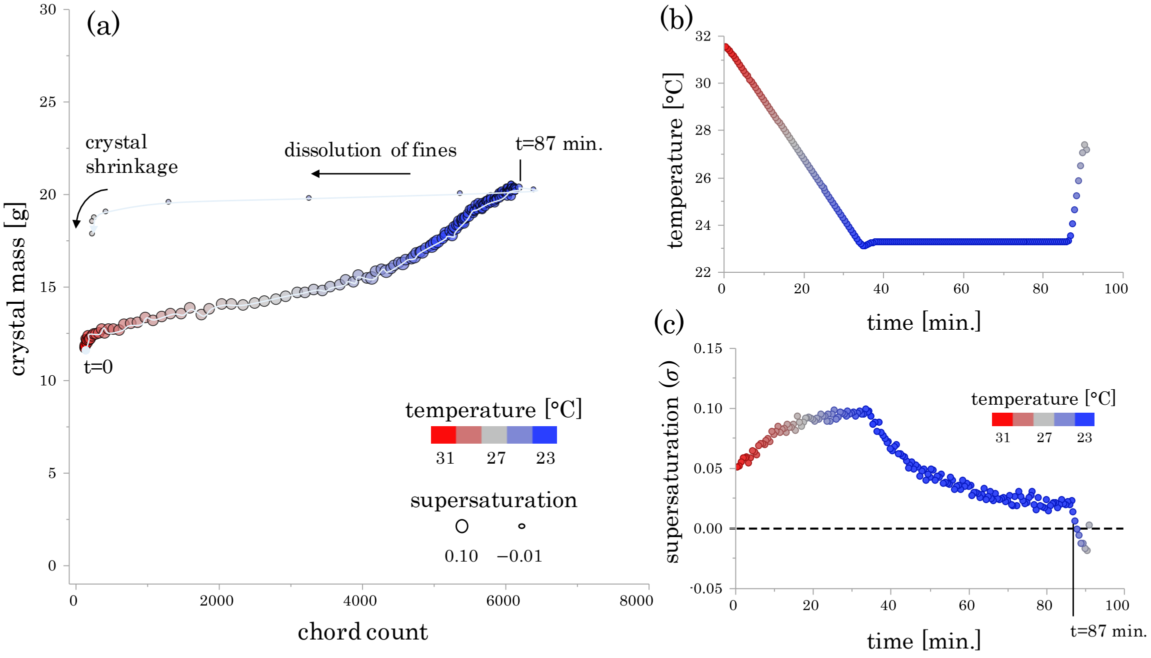

Figure 5.18: *Dissolution of fines.* (a) MC trajectory observed for paracetamol when fines are generated to add to an existing population of large crystals and then subsequently dissolved; (b) the temperature profile; (c) the supersaturation profile.

Complex dynamics can result in model-experiment mismatch. The complex dynamics observed for paracetamol associated with cooling-to-heating and heating-to-cooling switches, can result in a tortuous MC path. This may not be fully captured by the mathematical model which predicts the movement from just the current MC position and supersaturation, and is furthermore constrained to predict an increase in mass and count when the solution is supersaturated and, vice versa, a decrease in mass and count when the solution is undersaturated.

To provide a sense of the potential model-experiment mismatch, an example run is shown against the model prediction.

Figure 5.19 shows the MC trajectory for paracetamol when a series of cooling and heating cycles were implemented after reaching a high mass and count position. In this run, the MC trajectory moves up to the right and then spirals to the left—again, this type of trajectory is not fully captured by the model, as illustrated in Figure 5.20.

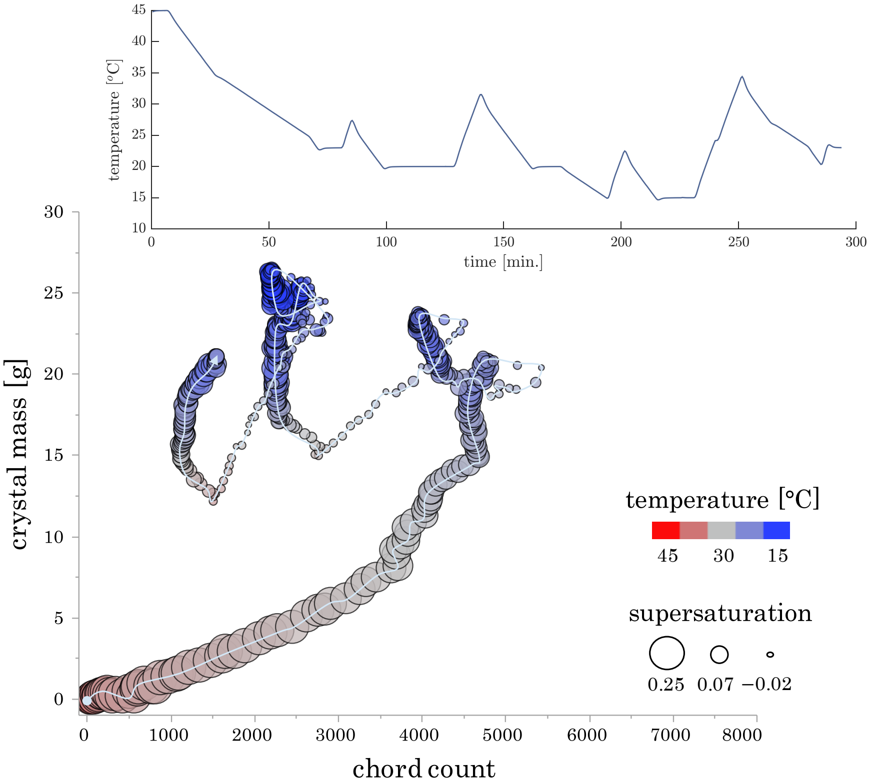

Figure 5.19: Complex MC trajectory observed for paracetamol crystallization and dissolution when temperature cycles are implemented from a high-mass, high-count position (positions indicated every 30 seconds). *Inset*: temperature profile.

For this same run, the MC trajectory was predicted from the temperature profile. The model-predicted MC trajectory is shown against the observed trajectory in Figure 5.20(a).

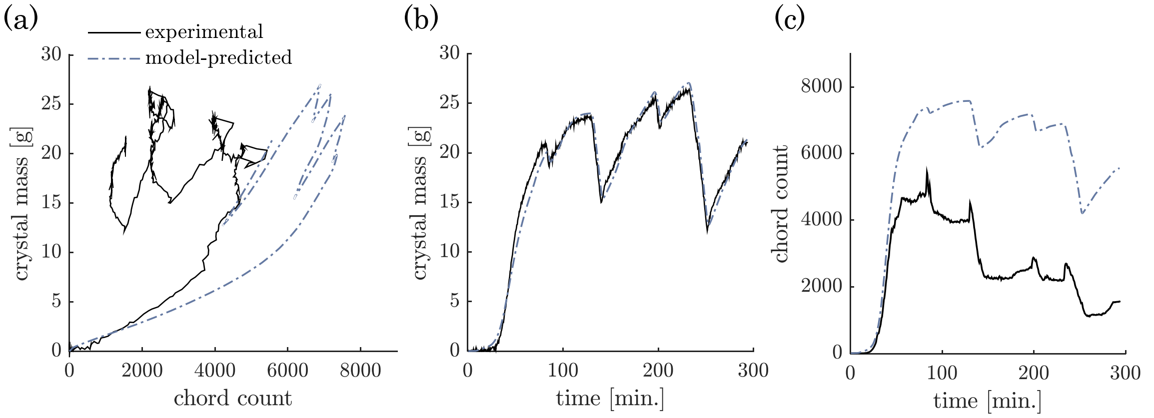

Figure 5.20: (a) Complex MC trajectory observed for paracetamol against the model-predicted trajectory for the same temperature input. (b) Measured crystal mass profile against the model-predicted profile. (c) Measured chord count profile against the model-predicted profile.

Separating out the mass profiles (Figure 5.20(b)) and count profiles (Figure 5.20(c)) we see where the model goes wrong: the predicted chord count profile deviates significantly from measured chord count profile. Deviations in the count predictions are typical for the complex paracetamol dynamics. As we will see in Chapter 6, this model-mismatch can degrade the effectiveness of model-based feedback control for the paracetamol system.

5.3 Chapter Conclusions

This chapter demonstrated application of the first step in the route to control within the MC framework: learning to move or drive in MC space.

A *conceptual* model of movement in MC space was first developed observing the different trajectories that resulted from implementation of various temperature profiles. Given as a set of rules, this conceptual model provides a general understanding that can be combined with feedback to establish control over the final MC position (as will be demonstrated in the next chapter). But this type of model does not admit precise calculation. Nor does it provide a clear distinction between the dynamics of the two different systems studied.

To overcome these drawbacks, we saw that machine learning could be applied to obtain a rigorous, mathematical model of the MC dynamics—specifically, a Markov state model. Here we found a difference in the dynamics of the two systems: the Markov state model of paracetamol crystallization indicates only a small effect of supersaturation on the angle of MC movement during crystallization, where as supersaturation was seen to have a large effect on the angle of MC movement during darapskite salt crystallization. This suggests that we have better control over darapskite salt crystallization than paracetamol crystallization. We also found that the obtained Markov state models do not fit the observed dynamics for the two systems equally well. In certain cases, the observed dynamics for paracetamol obviously deviated from the mathematical model. Nevertheless, both models capture the main effects and can therefore potentially be used to facilitate model-predictive feedback control over the MC position.

CHAPTER 6

CRYSTAL SIZE CONTROL

ONCE the dynamics are understood in terms of movement in MC space, strategies for driving to target locations—and thereby controlling the mean crystal size—can be developed. Feedback control policies are particularly powerful for this application.

In this chapter, the mass-count framework is used to develop feedback control policies, which are then applied to the studied experimental systems. The policies fall into two categories: rule-based policies and model-based policies.

The first type of policy—rule-based—is developed using the conceptual understanding of movement developed from observational learning in Chapter 5 (c.f. Figure 5.12). This understanding is translated to simple feedback rules expected to guide the MC trajectory towards the target from different locations in mass-count space.

The second type of policy—model-based—is developed with more precise calculation: the mathematical models of movement obtained from machine learning in Chapter 5 (c.f. figures 5.14 and 5.15) are used to calculate optimal feedback strategies. Both types of feedback strategies are shown to enable crystal size control.

6.1 Rule-Based Feedback Control

Feedback can often be leveraged to achieve remarkably good control over complex dynamical systems, even when the dynamics themselves are not fully understood or all that well characterized. We find the same to be true here. Given feedback on the crystal mass and chord count during the run, we can control the final mass-count position using only the most basic understanding of the dynamics.

Feedback interface. The MC framework sets up a natural interface for feedback control (c.f. §3.3.2). As shown again in Figure 6.1, the interface captures important information, including the relation between the current MC position and the target MC position. In combination with a general understanding of the movement, this information can be used to select the appropriate input action for moving towards the target.

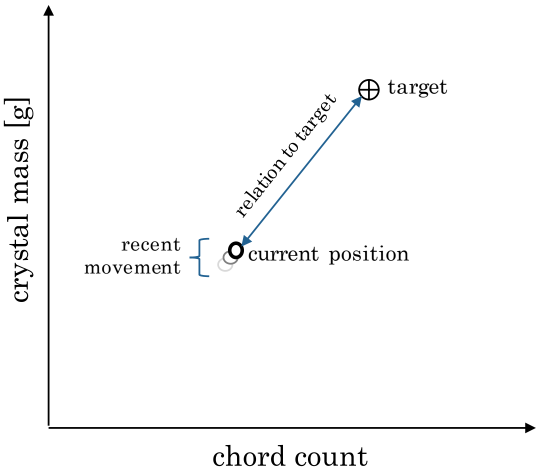

Figure 6.1: Visual interface for rule-based feedback control.

6.1.1 Fixed Zone Control

There are many ways to set up rule-based control. One simple way is what we call Fixed Zone Control. As the name is meant to suggest, this strategy defines operating rules based on fixed zones in mass-count space.

Fixed Zone Direct Nucleation Control (FZ-DNC). Figure 6.2 defines four zones around a target. The most basic understanding of movement in MC space suggests what we should do in Zones I and II to move towards the target: when we are in Zone I we should apply cooling, when we are in Zone II we should apply heating.

It is not quite as clear what should be done in Zones III and IV, but one option is to apply heating to come back towards the origin, into Zone I. This leads to the straightforward rule-based feedback control scheme defined in Table 6.1. This scheme is labeled Fixed Zone Direct Nucleation Control (FZ-DNC) in reference to a very similar, and already well-established, control scheme named Direct Nucleation Control (DNC) [Doki et al., 2004, Chew et al., 2007, Woo et al., 2009, Bakar et al., 2009b, Bakar et al., 2009a]. (The relation between the two schemes will be expanded in a later subsection).

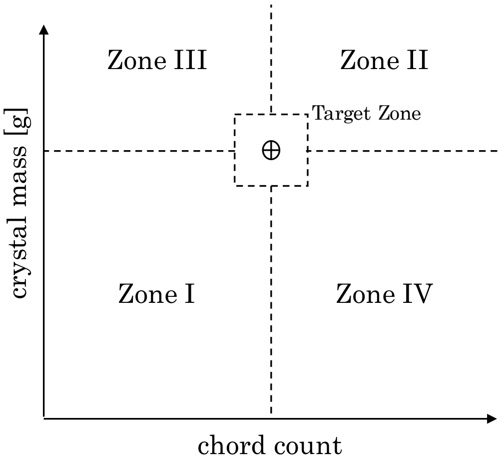

Figure 6.2: FZ-DNC Zones defined around a target position.

Table 6.1: Fixed Zone-Direct Nucleation Control (FZ-DNC) feedback algorithm.

```
Rule-Based Feedback Algorithm 1          FZ-DNC
repeat
    1. measure current mass and count
    2. if the mass count position is in Zone I*, cool
       else if the mass and count position is in Zone II, III, or IV, heat
until the Target Zone is reached
```

*Zones defined in Figure 6.2

FZ-DNC applied to paracetamol crystallization. The FZ-DNC control scheme was tested on paracetamol crystallization. The test strategy was simple and practical: we picked out three targets in mass-count space—which correspond to the production of crystals of three different sizes—and applied FZ-DNC to control the crystallization towards each of these. The targets selected for testing the control scheme are listed in Table 6.2.

Table 6.2: *Run targets.* FZ-DNC applied to paracetamol crystallization.

Run	**Target Position (count, mass [g])**	**Target Crystal Size mass-weighted mean [μm]**
1	(4000, 20)	223 (82–364)*
2	(1000, 20)	354 (211–497)
3	(275, 20)	544 (397–692)

*95% confidence interval based on mass-per-count–size correlation given in Figure 4.8

FZ-DNC control parameters. To apply the FZ-DNC scheme, the cooling and heating rates and the size of the Target Zone must be specified. We arbitrarily chose symmetric cooling and heating rates of −0.5°C/minute and 0.5 °C/minute; and defined the Target Zone to be the mass-count region within 10% of the target.

FZ-DNC feedback loop. With these control parameters set, the FZ-DNC scheme was applied using the cascade feedback loop shown in Figure 6.3. In the outer loop of the cascade, the mass-count position is fed to the controller, which in turn specifies the temperature setpoint based on the FZ-DNC algorithm. An internal proportional-integral (PI) loop is then used to adjust the temperature jacket to reach the temperature setpoint.

FZ-DNC results. Figure 6.4 shows the mass-count trajectory for paracetamol crystallization under FZ-DNC towards Target 2. In this case, the controller frequently switched between cooling and heating as the MC position moved between Zones I and IV. The net effect was the MC position ratcheting upwards towards the Target Zone, eventually reaching it after approximately 200 minutes.

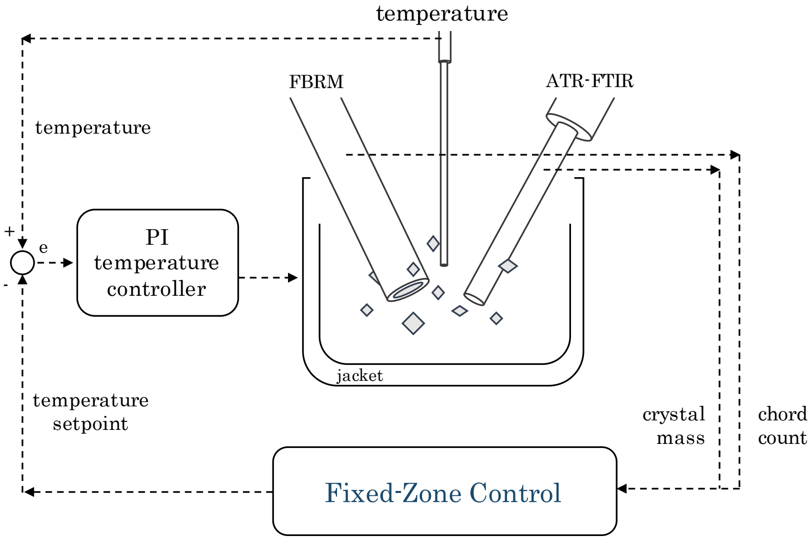

Figure 6.3: Schematic of the cascade feedback loop used to apply Fixed Zone feedback control.

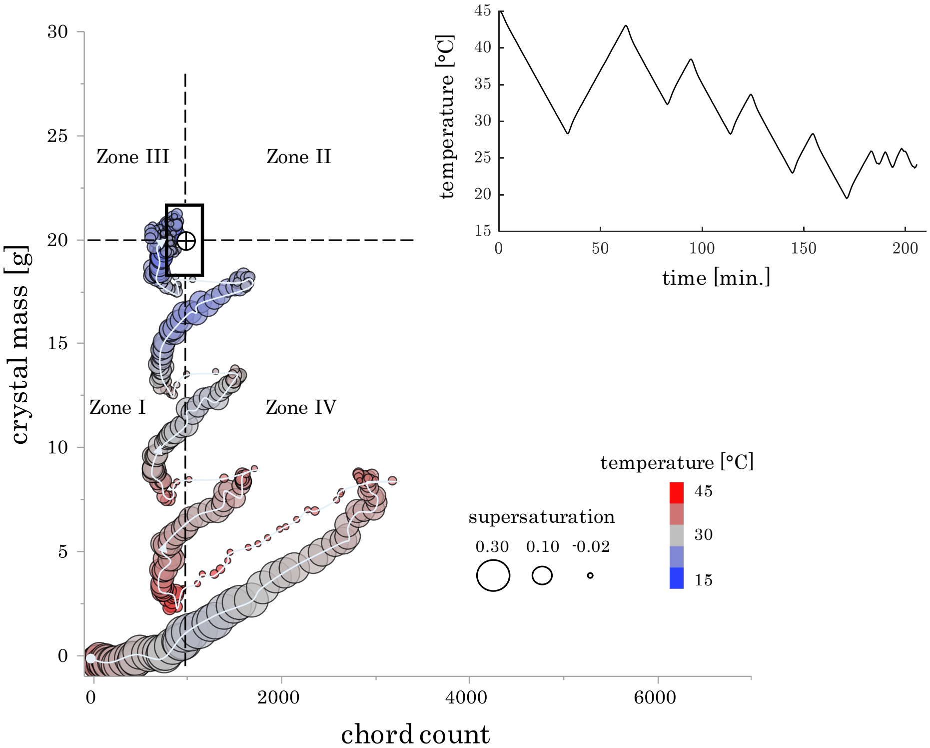

Figure 6.4: Mass-count trajectory realized in FZ-DNC Run 2 (positions indicated every 30 seconds). *Inset:* implemented temperature profile.

Figure 6.5 shows the trajectories for each of the three runs under FZ-DNC towards the three different targets. In every case, the feedback control scheme initiated linear cooling followed by heating-cooling cycles to ratchet the MC position upwards. This moved the MC position close to the target position in Runs 2 and 3. In Run 1, however, the controller did not move the MC position to the Target Zone and the run was manually ended with the MC position outside (to the left) of the Target Zone. Even so, application of the FZ-DNC scheme produced crystals of near target sizes:

- FZ-DNC Run 1 produced crystals with a measured mean size of 255 μm compared to a target of 223 μm;
- FZ-DNC Run 2 produced crystals with a measured mean size of 336 μm compared to a target of 354 μm; and,
- FZ-DNC Run 3 produced crystals with a measured mean size of 472 μm compared to a target of 544 μm.

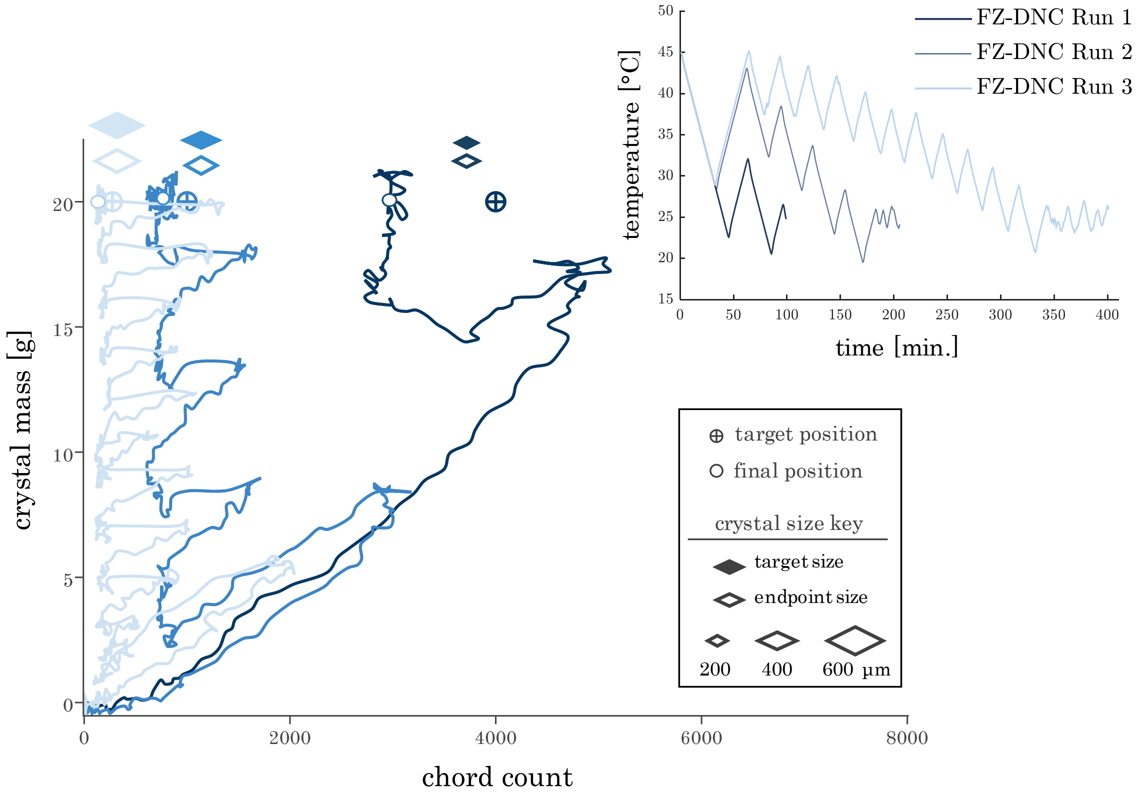

Figure 6.5: Summary of the mass-count targets and trajectories achieved with FZ-DNC. For each target, the measured mass-average crystal size is shown against the target size. *Inset:* implemented temperature profiles.

The measured crystal size results show that reasonably accurate size control is achieved with FZ-DNC (each measured crystal size is within the target size 95%

confidence interval). But, Run 1 also points out a weak spot in the controller: the run reached a position to the left of the target and then took little recourse to move the position back towards the target.

To substantiate that this is a consistent weakness of the control scheme and not just an artifact of the specific run, the FZ-DNC scheme was applied a second time towards Target 1. Figure 6.6 shows the results.

Again the controller guided the run to a position left of the target. Once such a position was reached, the controller implemented small temperature cycles as the MC position bounced back and forth between Zones I and III. These cycles did little to move the MC position to the right and so the run was again manually stopped at a position left of the Target Zone—again producing crystals with a mean size larger than the target size. As we will see in subsequent sections, the weakspot in the controller can be addressed by changing the zones and using slightly different rules.

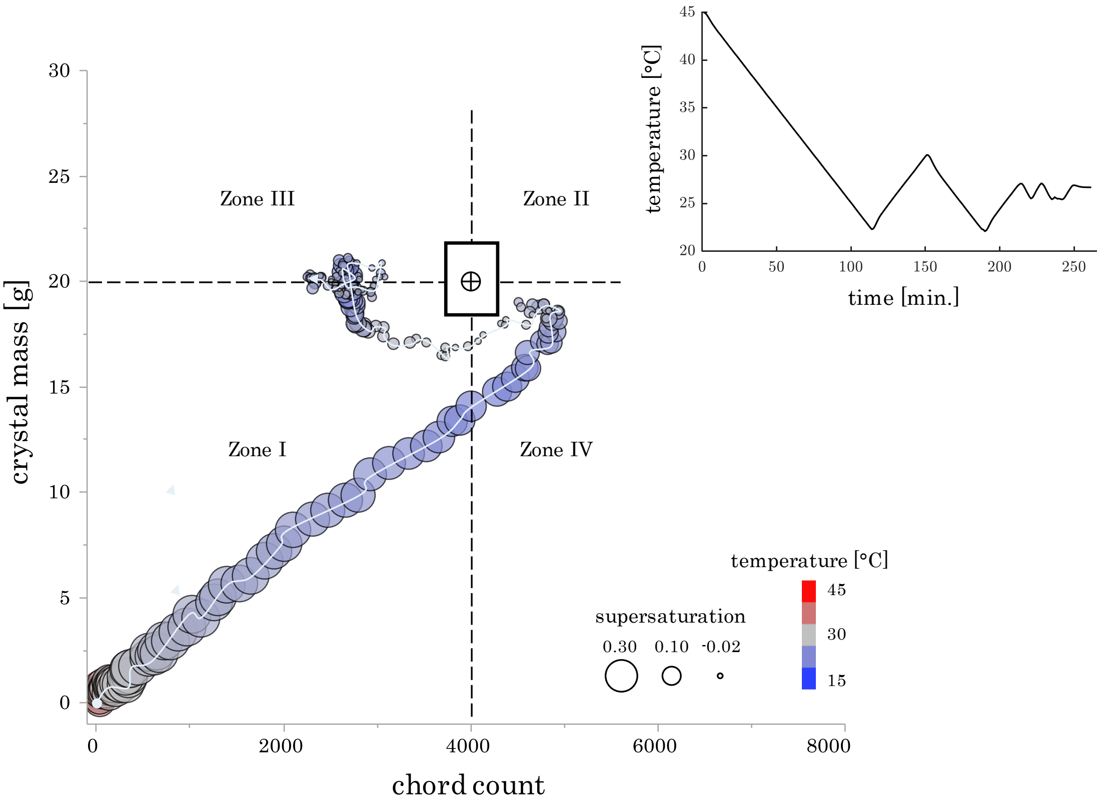

Figure 6.6: Mass-count trajectory realized when FZ-DNC was applied towards Target 1 a second time (positions indicated every 30 seconds). *Inset:* implemented temperature profile.

Relation to Direct Nucleation Control (DNC). The presented fixed zone control algorithm is labeled Fixed Zone *Direct Nucleation Control* to indicate its relation to, and inspiration from, the already well-established strategy referred to by nearly the same name: Direct Nucleation Control (DNC) [Doki et al., 2004, Chew et al., 2007, Woo et al., 2009, Bakar et al., 2009b, Bakar et al., 2009a].

In the most straightforward implementation of DNC, a chord count target line is picked; cooling is then applied when the count is below the target line and heating when the count is above the target line (the run ends when a set temperature is reached).

If the relation between mass-per-count and mean crystal size is known, DNC can be used to achieve approximately the same control as FZ-DNC. As an illustration, DNC was applied to paracetamol crystallization with target lines set at 4000 counts, 1000 counts, and 275 counts. In each case the runs where ended when the temperature reached 15 °C. The trajectories that resulted are shown in Figure 6.7.

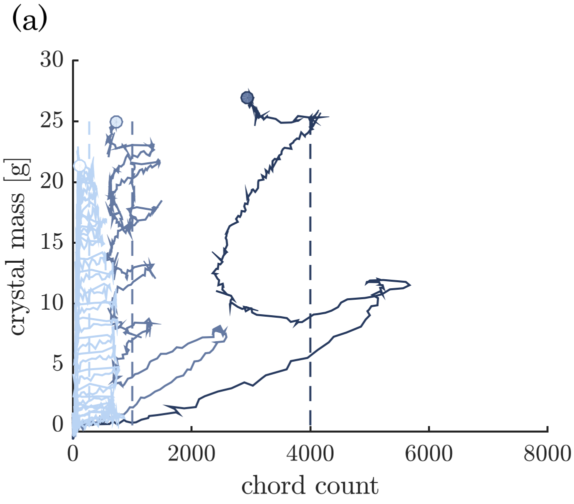

Figure 6.7: *DNC applied to paracetamol crystallization.* (a) MC trajectories for three paracetamol crystallizations under DNC with different count targets; (b) temperature profiles for the three runs.

Here we don't have a target position, just a chord count target line. Nevertheless, fixing the end temperature roughly controls the final crystal mass. DNC, as shown in Figure 6.7, can therefore be used to achieving approximate size control (again, provided the mass-per-count relationship is known ahead of time and the final temperature is appropriately adjusted).

But, as with FZ-DNC, control is predicated on the MC position ratcheting upwards around the target count line and the controller has little recourse if this does not happen. More accurate and potentially more efficient control can be achieved by applying schemes that utilize a more elaborate understanding of movement in mass-count space and also leverage feedback on *both* the chord count and crystal *mass*.

An alternative fixed zone strategy: Fixed Zone adjusted Direct Nucleation Control (FZ-aDNC). The zones used for FZ-DNC are not the only zones we can define. Nor are they necessarily the most useful. Here we define a slightly adjusted Fixed Zone Control strategy aimed at achieving more accurate control and reaching the target position more efficiently. The new zones are shown in Figure 6.8.

Instead of defining the zones to be the four quadrants around the target, three triangular zones are defined. These zones are selected considering the conceptual understanding of MC movement for paracetamol crystallization and dissolution gained in Chapter 5, taking into account the movement observed on the switch from heating to cooling at high-mass, high-count positions (c.f Figure 5.17). Consequently, these zones can be used to set up more reliable control.

The operation we propose using the zones in Figure 6.8 is straightforward: in Zone I the solution is cooled quickly; in Zone II the solution is cooled slowly; and, finally, in Zone III the solution is heated. The temperature is fixed and the run ends when the Target Zone is reached. This algorithm, labeled FZ-aDNC, is given in Table 6.3.

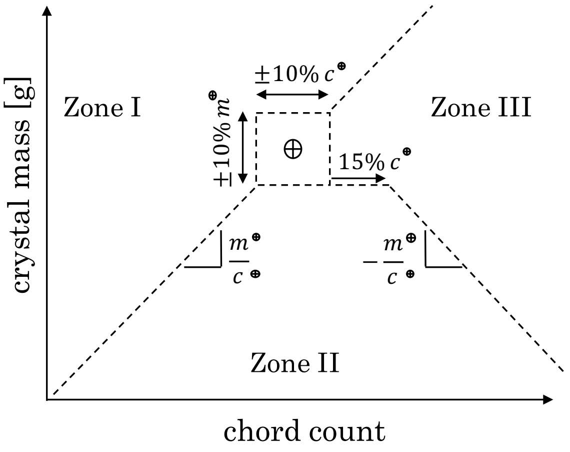

Figure 6.8: FZ-aDNC Zones defined around a target position.

FZ-aDNC control parameters and feedback loop. Cooling and heating rates must be pre-specified to operate the FZ-aDNC strategy. In application, the fast cooling rate was arbitrarily set to −1.0 °C/minute, the slow cooling rate set to −0.5 °C/minute, and the heating rate set to 0.5 °C/minute. The same cascade feedback

Table 6.3: Fixed Zone adjusted Direct Nucleation Control (FZ-aDNC) algorithm.

```
Rule-Based Feedback Algorithm 2        FZ-aDNC
repeat
        1.  measure current mass and count
        2.  if the mass count position is in Zone I*, cool quickly
            else if the mass and count position is in Zone II, cool slowly
            else if the mass and count position is in Zone III, heat
until $t > t_N$ or the Target Zone is reached
```

*Zones defined in Figure 6.8

control loop shown in Figure 6.3 was then used to apply FZ-aDNC.

FZ-aDNC applied to paracetamol crystallization. The FZ-aDNC strategy is tested on paracetamol crystallization for the same set of three targets given in Table 6.2.

Figure 6.9 shows the trajectory achieved under FZ-aDNC towards the first target. Here, we see that the FZ-aDNC algorithm is able to quickly reach a position close to the target. Compared to the FZ-DNC, the adjusted FZ-aDNC scheme appears to be more efficient and more accurate.

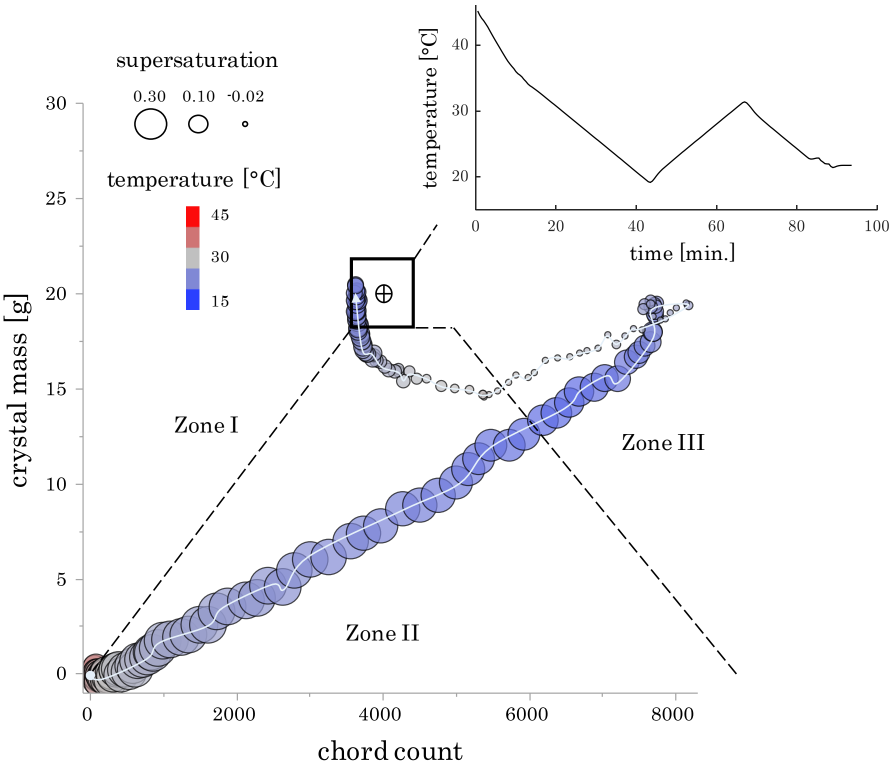

Figure 6.9: Mass-count trajectory realized in FZ-aDNC Run 1 (positions indicated every 30 seconds). *Inset:* implemented temperature profile.

The three trajectories under FZ-aDNC towards the selected targets are shown together in Figure 6.10. In each case the scheme controlled the MC trajectory to reach a position near the target. This resulted in the production of crystals of near the target sizes:

- FZ-aDNC Run 1 produced crystals with a measured mean size of 259 μm compared to a target of 223 μm;
- FZ-aDNC Run 2 produced crystals with a measured mean size of 323 μm compared to a target of 354 μm; and,
- FZ-aDNC Run 3 produced crystals with a measured mean size of 577 μm compared to a target of 544 μm.

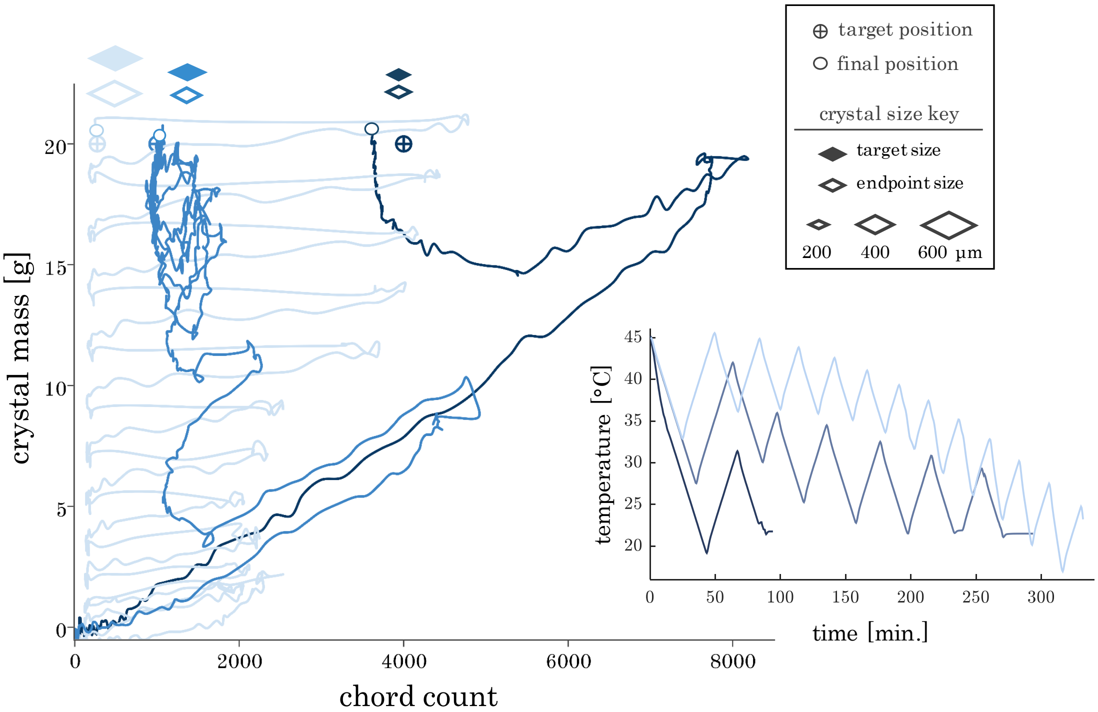

Figure 6.10: Summary of the mass-count targets and trajectories achieved with FZ-aDNC. For each target, the measured mass-average crystal size is shown against the target size. *Inset:* implemented temperature profiles.

Fixed Zone Control schemes are susceptible to high-frequency switching and endless loops. Fixed-Zone (FZ) control schemes are simple—both to develop and to implement. They also proved to be reasonably effective in the examples given. But the rigidity of these schemes can have a drawback: Fixed Zone schemes are susceptible to high-frequency switching and looping. Avoidance of these traps is reliant on having appropriately delayed actuation and the system having an asymmetric response to heating and cooling or showing hysteresis (or both). Let me expand on this.

First, high-frequency switching: when the MC position moves from one zone to another (for example, from Zone I to Zone IV in Figure 6.2), the controller may fall into a trap, quickly switching between heating and cooling as the position moves back and forth rapidly between zones. To avoid this type of rapid switching, there must be some delay in the actuation—conferred by a slow sampling time or a slow temperature change response time or a delay in the dynamic response of the system.

As an example, consider the DNC action (as applied here). When the count is below a threshold line, cooling is applied; when the count is above the threshold, heating is applied. Cooling is expected to increase the count and heating is expected to decrease the count—the control scheme would make no sense otherwise. But if these effects were enacted *immediately* then the control would switch rapidly between heating and cooling as the count oscillated around the threshold line.

And high-frequency switching only one facet of the problem. Even if rapid switching is avoided by delayed actuation, to make progress using Fixed Zone Control also requires the system to also have an asymmetric response to heating and cooling or to show clear hysteresis—the time-based dependence of the systems response to an input. That is, if the MC movement caused by heating is always just the mirror opposite of the movement caused by cooling, then applying cooling-heating cycles—like those enacted by the Fixed Zone schemes—would simply cause the MC position to move back and forth, retracing the earlier trajectory. The control would fall into an endless loop.

The problem of high-frequency switching and looping was seen in the some of the FZ-DNC control runs when the target was missed. In particular, if the MC position reached a position to the left of the target zone, cooling was applied when the mass was below the target threshold line and heating when it was above the threshold. The dynamic response of mass to temperature change (in this region) was reasonably quick. As a result, the controller quickly switched between cooling and heating. Moreover, the effect of heating was to roughly retrace the path taken by cooling and so no progress was made.

Adaptive Zone schemes, as will be described in the next section, can be designed to avoid high-frequency switching and reduce the risk of looping.

6.1.2 Adaptive Zone Control

Here we develop a slightly more sophisticated rule-based scheme in which the zones shift depending on the operating mode and, also, details of relation between the current position and target are used within a zone to tailor the controller input. The scheme is termed spatially-Guided Action, Trajectory Endpoint Control (sGATEC) [Griffin et al., 2015c].

The sGATEC scheme operates in two modes: Cooling Mode and Heating Mode. Given the mode of operation, the spatial zones shown in Figure 6.11 are used to select the appropriate temperature input as indicated in Table 6.4.

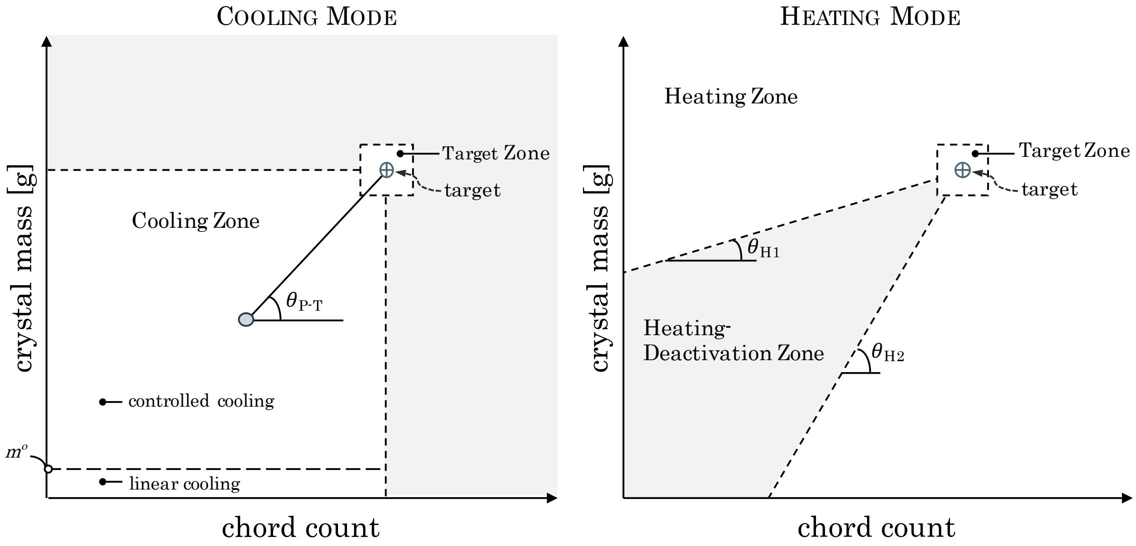

Figure 6.11: Spatial zones used to guide actions of the sGATEC rule-based feedback control scheme.

The sGATEC scheme is somewhat involved, but becomes clear with a step-by-step explanation. Starting in Cooling Mode, the sGATEC scheme works as follows:

- **Cooling Mode.** In Cooling Mode, the temperature is decreased. If the crystal mass is below a select threshold (m^0 in Figure 6.11), the temperature is decreased linearly. Alternatively, if the crystal mass is above this threshold value (but still within the cooling zone) the temperature is decreased in such a way that the supersaturation is maintained at a desired setpoint. This setpoint is selected according to the angle between the current position and the target position, $\theta_{\text{P-T}}$ (lower supersaturation setpoints are used for higher angles).

- **Switch from Cooling Mode to Heating Mode.** The mode switches from Cooling Mode to Heating Mode if the MC position moves outside the Cooling Zone and not into the Target Zone.
- **Heating Mode.** In Heating Mode, the operation is simple: linear heating is applied.
- **Switch from Heating Mode to Cooling Mode.** If heating moves the MC position out of the Heating Zone, but not into the Target Zone, Cooling Mode in re-enacted.

The operation continues switching between Cooling Mode and Heating Mode unless the MC position moves into the Target Zone. Once the Target Zone in reached, the temperature is fixed and the run ends.

Table 6.4: sGATEC feedback algorithm.

```
Rule-Based Feedback Algorithm 3        sGATEC
start in cooling mode
Cooling Mode
    1.  measure current mass and count
    2.  if the mass and count are in the Target Zone*,
            hold temperature and end run
        else if, the mass is below the threshold $m^0$,
            apply linear cooling
        else if, the mass and count are in the Cooling Zone,
            apply supersaturation control with the setpoint
            determined by the angle $\theta_{\text{P-T}}$
        else, switch to Heating Mode
Heating Mode
    1.  measure current mass and count
    2.  if, the mass and count are in the Target Zone,
            hold temperature and end run
        else if, the mass and count are in the Heating Zone,
            apply linear heating
        else, switch to Cooling Mode
```

*Zones defined in Figure 6.11

sGATEC control parameters. A number of control parameters must be specified to apply the sGATEC scheme in practice, including: the mass threshold, the zone angles, and the heating and cooling rates. The control parameters used in applying the sGATEC scheme to darapskite salt crystallization were developed in [Griffin et al., 2015c]. These are listed in Appendix §D.1.

sGATEC feedback loop. Once the control parameters were defined, the sGATEC control policy was applied with the cascade feedback control loop shown in Figure 6.12. This feedback loop is very similar to that used for Fixed Zone Control, with one slight difference: the solute concentration as well as the solubility is needed at each measurement interval to infer the temperature setpoint from the supersaturation setpoint output by the sGATEC controller.

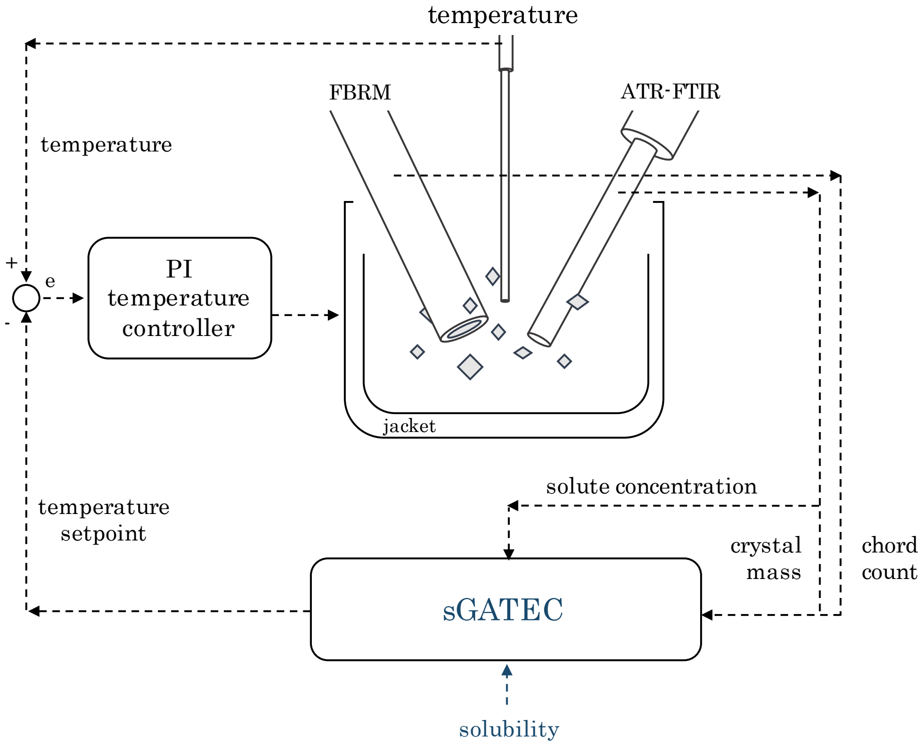

Figure 6.12: Schematic of the cascade feedback loop used to apply the sGATEC control policy.

sGATEC applied to darapskite crystallization. The sGATEC scheme was tested on darapskite salt crystallization [Griffin et al., 2015c]. As with the other tests, three targets were picked in mass-count space, which correspond to the production of crystals of three different sizes. These are listed in Table 6.5.

Table 6.5: *Run targets.* sGATEC control applied to darapskite system.

Run	Target Position (count, mass [g])	Target Crystal Size mass-weighted mean [μm]
1	(295, 7)	422 (213–643)*
2	(210, 12)	566 (357–790)
3	(75, 13)	819 (611–1050)

* 95% confidence interval based on mass-per-count–size correlation given in Figure 4.7

Figure 6.13 shows the trajectory observed for darapskite salt crystallization under the sGATEC scheme towards the second target. This highlights the action of sGATEC: the controller automatically applies what can be interpreted as "internal seeding" followed by adaptive supersaturation control to reach the target position.

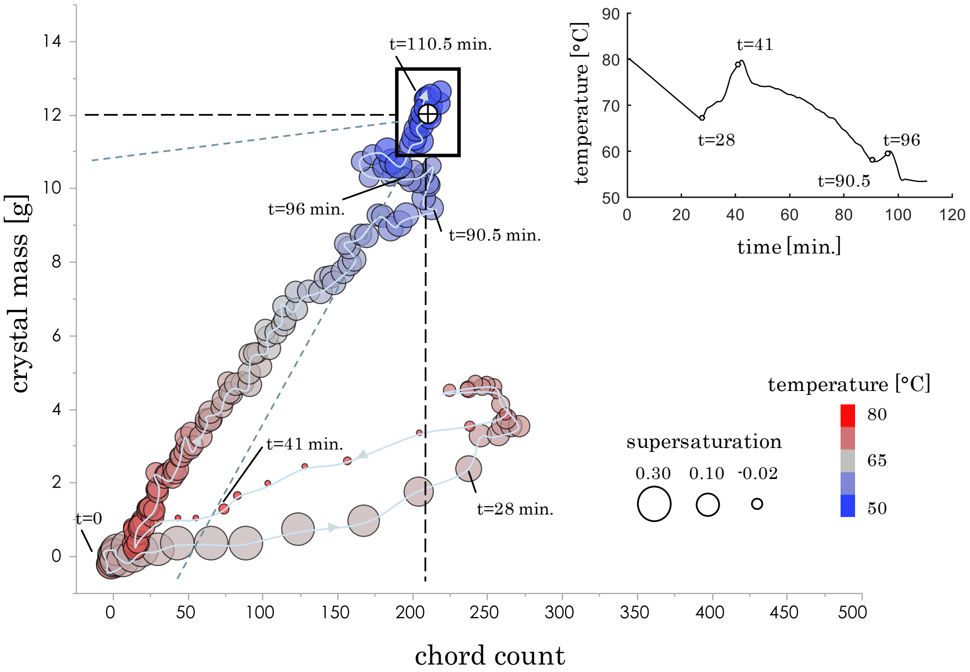

Figure 6.13: Mass-count trajectory realized in sGATEC Run 2 (positions indicated every 30 seconds). *Inset:* implemented temperature profile.

Figure 6.14 shows the trajectories observed under sGATEC when implemented towards the three different targets. In each case, the sGATEC scheme varied the temperature profile—again, applying supersaturation control and temperature cycles—to produce trajectories that ended near the target positions. This resulted in the production of crystals with mean sizes near the target sizes:

- sGATEC Run 1 produced crystals with a measured mass-weighted mean size of 460 μm compared to a target of 422 μm;
- sGATEC Run 2 produced crystals with a measured mass-weighted mean size of 671 μm compared to a target of 566 μm; and,
- sGATEC Run 3 produced crystals with a measured mass-weighted mean size of 915 μm compared to a target of 819 μm.

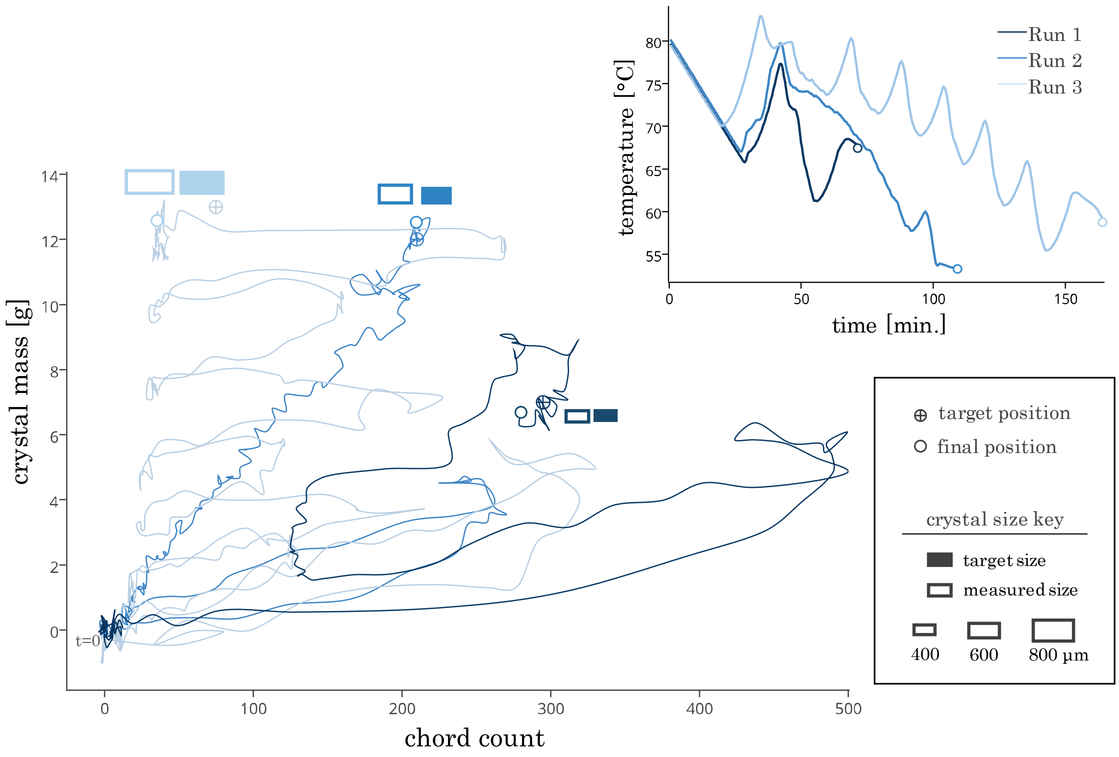

Figure 6.14: Summary of the mass-count targets and trajectories achieved with sGATEC. For each target, the measured mass-average crystal size is shown against the target size. *Inset:* implemented temperature profiles.

6.1.3 Summary of rule-based control

The MC framework yields a useful feedback interface that facilitates the development of rule-based schemes. Of the many possible rule-based schemes we could think of, three were developed and demonstrated here: FZ-DNC, FZ-aDNC, and sGATEC. The first scheme, FZ-DNC, is a fixed zone scheme that operates in a manner similar to Direct Nucleation Control. The second scheme, FZ-aDNC, is a variation of the first—using different zones to generate feedback rules more in-tune with the observed dynamics for paracetamol. The last scheme, sGATEC, is slightly more involved. It uses zones that shift with operating mode and also selectively applies supersaturation control. Figure 6.15 summarizes the experimental results obtained for these rule-based schemes.

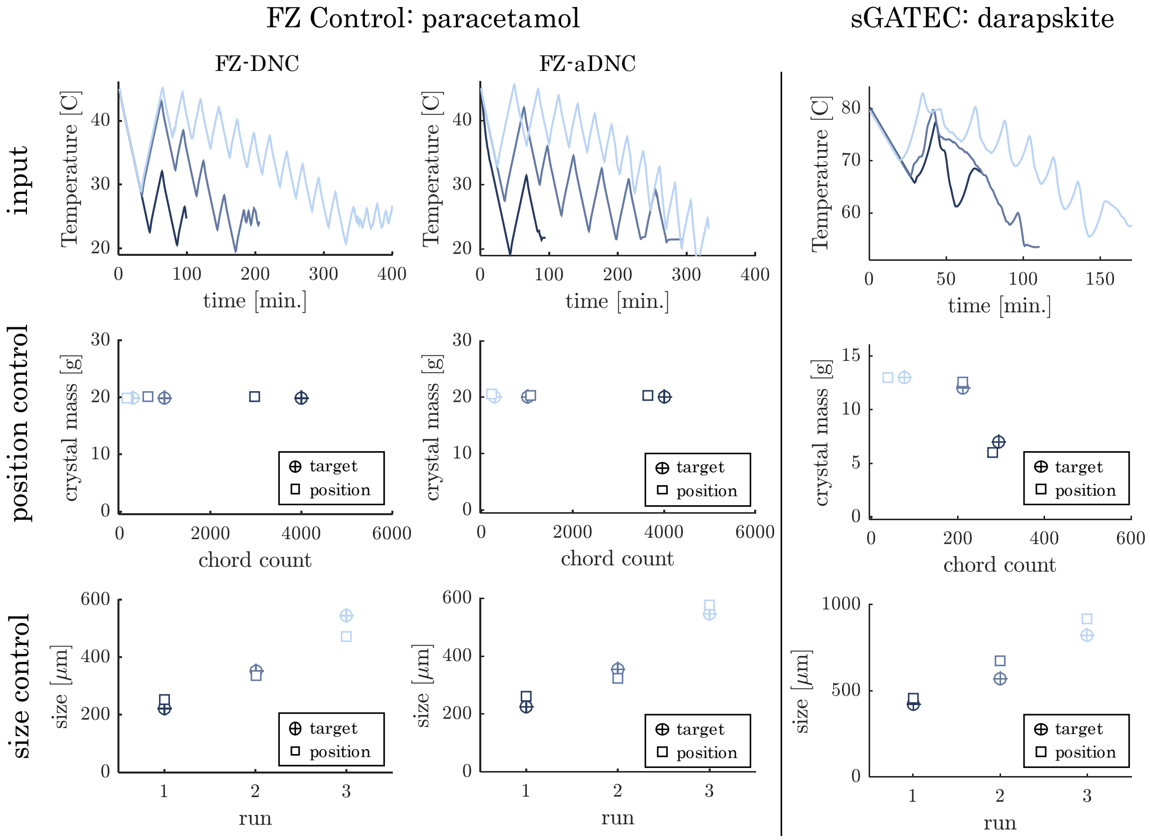

Figure 6.15: Summary of the control achieved with rule-based feedback control.

From this figure we see the power of feedback control. Built from a purely conceptual understanding of the dynamics, the rule-based feedback schemes are effective for achieving quantitative control: implementing complex temperature profiles to produce crystals of near target sizes for both systems.

Notwithstanding the success of the rule-based feedback schemes for producing crystals of targeted sizes, there are drawbacks to model-free control. The batch times are not know *a priori* and the optimality of the schemes cannot really be evaluated. For this we need to utilize model-based control.

6.2 Model-Based Feedback Control

The MC space Markov state model of crystallization and dissolution dynamics given by Equation (5.1) can be used to identify optimal feedback control policies. That is, we can ask: "what is the best policy for reaching a target position, provided that the dynamics behave exactly according to the mathematical model?" And then we can use computational mathematics to come up with the answer.

6.2.1 Optimal control formulation

The optimization problem associated with state-feedback control has the form:

$$\begin{array}{ll} \underset{u_\tau = \pi(\mathbf{x}_\tau)}{\text{minimize}} & \Phi\left(\mathbf{x}_{\tau=1,\ldots,N}, u_{\tau=1,\ldots,N}\right) \\ \text{subject to} & \mathbf{x}_{\tau+1} = F\big(\mathbf{x}_\tau, u_\tau = \pi(\mathbf{x}_\tau)\big)\Delta t + \mathbf{x}_\tau,\ \tau = 0,\ldots,N-1; \\ & \mathbf{x}_0 = \mathbf{x}^{\text{init}}; \end{array}$$

where $\Phi : \mathbf{R}^{d+1,N} \mapsto \mathbf{R}$ is the control *objective function* and $\pi : \mathbf{R}^d \mapsto \mathbf{R}$ is the *state-feedback control policy.*

Control objective function. We aim to produce crystals of select mean sizes in set batch times. From the mass-count perspective, this primary control objective is represented by a target mass-count position, $\mathbf{x}^{\oplus}$, to be reached in a fixed batch time,

t_N. But there is an additional consideration: we would also like to operate efficiently, with minimal input energy. To express this multifaceted objective, two functions are defined:

- an *input-effort* function
$$\varepsilon(u_\tau) \equiv u_\tau^2; \text{ and}$$
- a *distance-to-target* function
$$d(\mathbf{x}_\tau, \mathbf{x}^\oplus) \equiv (\mathbf{x}_\tau - \mathbf{x}^\oplus))^\top Q(\mathbf{x}_\tau - \mathbf{x}^\oplus), \text{ where } Q = \begin{bmatrix} 1 & 0 \\ 0 & \lambda^2 \end{bmatrix}.$$

The input-effort function reflects the following: crystallization or dissolution will spontaneously bring the system to saturation; maintaining undersaturation or supersaturation during the operation requires a continual temperature change and, consequently, the input of energy.

The distance-to-target function corresponds to the normalized, squared Euclidean distance between the mass-count position and the target position. In this measure, we scale the chord count by λ (g/#) in order to provide approximately equal weight to the chord count (#) and mass measurements (g).

Using these two functions we can write a weighted objective function:

$$\Phi\left(\mathbf{x}_{\tau=1,\ldots,N}, u_{\tau=1,\ldots,N}\right) = \overbrace{\rho}^{\text{weight}} \underbrace{\sum_{\tau=0}^{N-1} \varepsilon(u_\tau)}_{\text{input energy}} + \underbrace{d(\mathbf{x}_N, \mathbf{x}^\oplus)}_{\text{final distance from target}} \tag{6.1}$$

The optimization problem that minimizes this objective function roughly reads: "find the optimal policy for bringing the system to the target position in the given batch time with minimal energy input."

This aligns with the conceptual objectives already specified, but it still misses at least one practical consideration, which is that we want the system to be settled at the end of the run—i.e. we do not want the crystal mass and chord count to be transient at the end of the batch.

In order to preferentially select trajectories that move towards the target as directly as possible and then settle, the time-varying term $(t_\tau/t_N)^\gamma d(\mathbf{x}_\tau, \mathbf{x}^\oplus)$ is added to the running cost. This gives us a final objective function:

$$\Phi = \underbrace{\rho \sum_{\tau=0}^{N-1} \left\{(t_\tau/t_N)^\gamma d(\mathbf{x}_\tau, \mathbf{x}^\oplus) + \varepsilon(u_\tau)\right\}}_{\text{running cost}} + \underbrace{d(\mathbf{x}_N, \mathbf{x}^\oplus)}_{\text{terminal cost}}\ , \tag{6.2}$$

and a final optimization problem:

$$\begin{array}{ll} \underset{u_\tau = \pi(\mathbf{x})}{\text{minimize}} & \rho \sum_{\tau=0}^{N-1} \left\{(t_\tau/t_N)^\gamma d(\mathbf{x}_\tau, \mathbf{x}^\oplus) + \varepsilon(u_\tau)\right\} + d(\mathbf{x}_N, \mathbf{x}^\oplus) \\ \text{subject to} & \mathbf{x}_{\tau+1} = F\big(\mathbf{x}_\tau, u_\tau = \pi(\mathbf{x}_\tau)\big)\Delta t + \mathbf{x}_\tau,\ \tau = 0, \ldots, N-1, \\ & \mathbf{x}_0 = \mathbf{x}^{\text{init}}. \end{array} \tag{6.3}$$

Applying dynamic programming to solve the optimization problem. The optimization problem (6.3) is non-convex (F is non-linear), making it generally difficult to solve [Boyd and Vandenberghe, 2009], especially for high-dimensional systems. Here we see another advantage of having a low-dimensional model of the dynamics: for a two-dimensional state, the state space can be discretized and dynamic programming can then be applied to solve the state-feedback optimization problem. This is described in detail in Appendix §D.2.

The state-feedback control policy. The solution to the optimization problem (6.3) is denoted π^*. This is the optimal state-feedback control policy that specifies the supersaturation setpoint as a function of the current MC position and time:

$$\pi^* : \mathbf{x}_\tau \mapsto u_\tau^*.$$

This solution will vary with a number of variables that the user chooses. These include: the target position, $\mathbf{x}^\oplus$; the final batch time, denoted by the index N; and the optimization parameters, ρ and γ.

Once obtained, the state-feedback policy can be represented simply as a lookup table that gives the appropriate input (supersaturation setpoint) to apply for each different position, $\mathbf{x}$, in the discretized space at each different time interval. To give a sense of these policies, the lookup tables can be displayed as color maps that show the policy-suggested inputs by color. Figures 6.17 and 6.20, for example, display obtained optimal control policies in this way.

The state-feedback control policy enacts *explicit* model predictive control (MPC). In many fields, Chemical Engineering among them, a strategy labeled model-predictive control (MPC) is the standard-bearer for model-based feedback control. The conventional MPC strategy is as follows. At each time interval, the current state is measured, $\hat{\mathbf{x}}_t$, and a finite-horizon optimal control problem is posed using a model of the dynamics:

$$\begin{aligned} \underset{u_t \ldots u_{\mathcal{T}-1}}{\text{minimize}} \quad & \sum_{\tau=t}^{\mathcal{T}-1} \Big(\phi\left(\mathbf{x}_\tau, u_\tau\right)\Big) \\ \text{subject to} \quad & \mathbf{x}_{\tau+1} = F\big(\mathbf{x}_\tau, u_\tau = \pi(\mathbf{x}_\tau)\big)\Delta t + \mathbf{x}_\tau, \ \tau = t, \ldots, \mathcal{T}-1; \\ & \mathbf{x}_t = \hat{\mathbf{x}}_t; \end{aligned}$$

where ϕ is the running cost over the time horizon, $t, \ldots, \mathcal{T}-1$.

The solution to this optimization problem is the optimal open-loop input profile, $u_t^*, \ldots, u_{\mathcal{T}-1}^*$. In application, the first input is applied and the process is repeated: with each new measured state, the fixed-horizon control problem is re-solved and the first input of the new solution sequence is applied. This, in effect, is a state-feedback control strategy, $u_\tau^* = \pi_{\text{mpc}}(\hat{\mathbf{x}}_\tau)$.

So how does the MPC strategy relate to the state-feedback control policy we obtain from solving the optimization problem (6.3)? They are the same, with two qualifications: the time horizon, in our application, always extends from the current time to the specified end-of-run time; and the feedback control input is found *ahead of time* for *each different position* in the discretized space at *each time interval*. The strategy we apply is sometimes referred to as *explicit* model predictive control [Alessio and Bemporad, 2009].

6.2.2 Applying the state-feedback control policy

Optimal state-feedback control policies, π^*, obtained offline through dynamic programming were applied online using a cascade feedback loop similar to that applied for the rule-based control schemes. The cascade feedback loop is shown in Figure 6.16. In the outer loop, the mass-count position is fed to the state-feedback controller, which uses the lookup table representation of $\pi^*(\mathbf{x}_\tau)$ to identify the appropriate supersaturation setpoint. The inner loop then adjust the temperature to hit this setpoint.

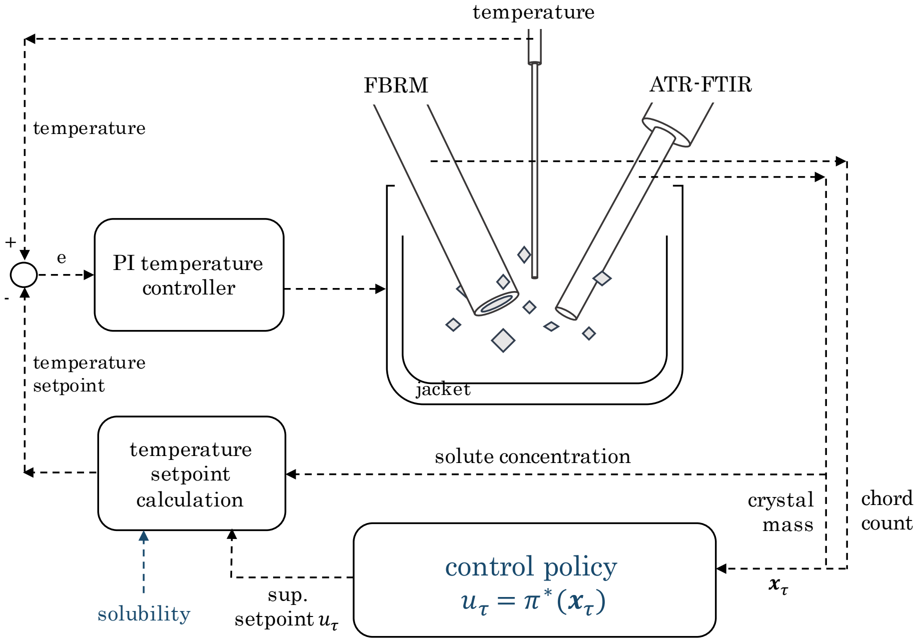

Figure 6.16: Schematic of the cascade feedback loop used to apply optimal state-feedback control policies.

6.2.3 Application to darapskite crystallization

The model-based feedback control strategy was tested experimentally on darapskite salt crystallization. As was done to test the rule-based schemes, three targets were selected. In this case, the target batch times were also set, with run times selected to be near optimal for the given targets (c.f. §C.2.2). Table 6.6 summarizes the selected targets and times.

Table 6.6: *Run targets.* Model-based control applied to darapskite system.

Run	**Target Position (count, mass [g])**	**Target Crystal Size mass-weighted mean [μm]**	**Target Batch Time [min.]**
1	(300, 7)	420 (210–641)*	30
2	(200, 10)	541 (332–765)	60
3	(75, 11)	775 (566–1004)	120

$*$ 95% confidence interval based on mass-per-count–size correlation given in Figure 4.7

Control policies. For each given target and batch time, the optimization problem (6.3) yields a different control policy. To inspect these policies, they can be visualized as time-varying color maps that show the suggested input for each MC position at different time intervals. Figures 6.17 and 6.18 show the optimal state-feedback control policies obtained for Test Runs 1 and 2, respectively.

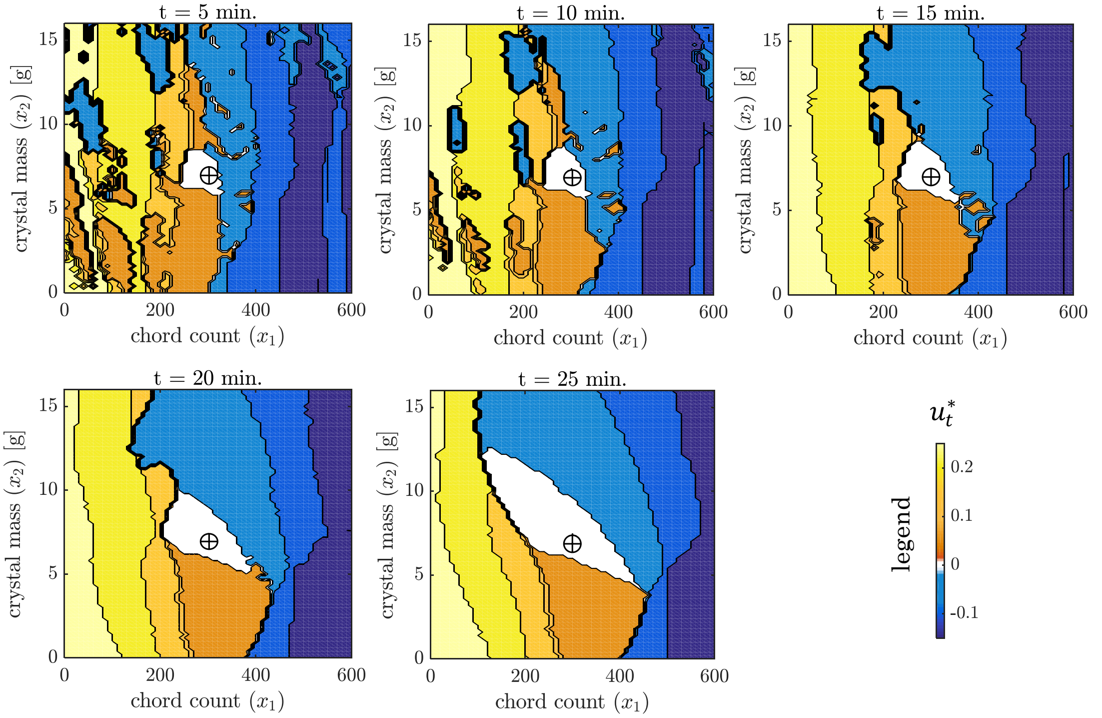

Figure 6.17: Color maps of the optimal state-feedback policy, π, for darapskite crystallization towards Target 1 shown at representative times during the process (during the control run the map updates every 5 minutes).

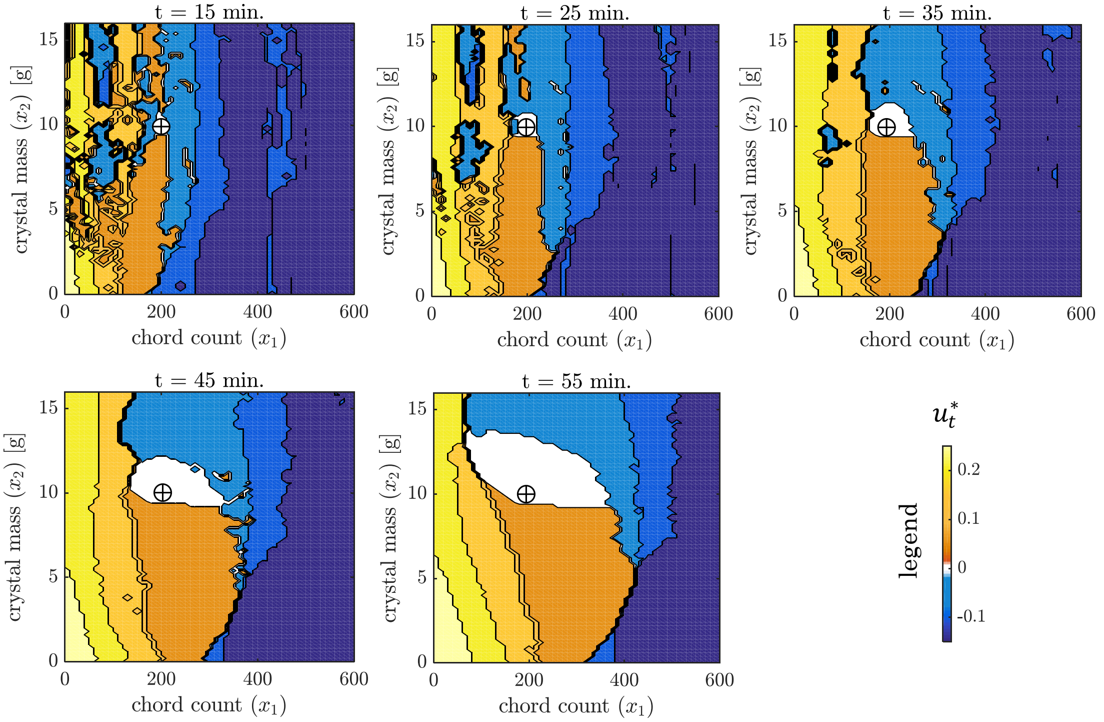

Figure 6.18: Color maps of the optimal state-feedback policy, π, for darapskite crystallization towards Target 2 shown at representative times during the process (during the control run the map updates every 5 minutes).

In displaying the optimal state-feedback control policies we see that they have many similarities to the rule-based schemes developed in the previous section. Below and to the left of the target, the policies generally dictate the application of a positive supersaturation setpoint (achieved by cooling). Above and to the right of the targets the policies suggest the opposite: application of a negative supersaturation setpoint (achieved by heating).

But the model-based policies differ from the rule-based control policies in the details. For example, as the position approaches the target, the suggested supersaturation (undersaturation) levels are more tempered. Moreover, complex series of actions are suggested for positions below and to the right or above and to the left of the targets. These details show foresight: tempering the supersaturation near the target slows down the movement and combats overshoot; complex actions are required to move the position net up and left or net down and right.

The model-based policies also exhibit foresight in knowing when to quit. The white region around the target can be thought of as an expanding target zone: If the MC position happens to move into this zone, the policy suggests that any remaining actions in the limited batch time remaining will be detrimental—moving the MC position farther from the target.

Model-based control of darapskite crystallization. Figure 6.19 gives a summary of the experimental results for model-based control applied to darapskite crystallization. In each of the test runs, the optimal state-feedback controller guided the MC trajectory to reach a position near the target position in the pre-specified batch time through the application of a unique temperature profile. This resulted in the production of crystals of near target sizes:

- Run 1 under optimal control produced crystals with a measured mass-weighted mean size of 386 μm compared to a target of 420 μm;
- Run 2 under optimal control produced crystals with a measured mass-weighted

mean size of 517 μm compared to a target of 566 μm; and,

- Run 3 under optimal control produced crystals with a measured mass-weighted mean size of 731 μm compared to a target of 775 μm.

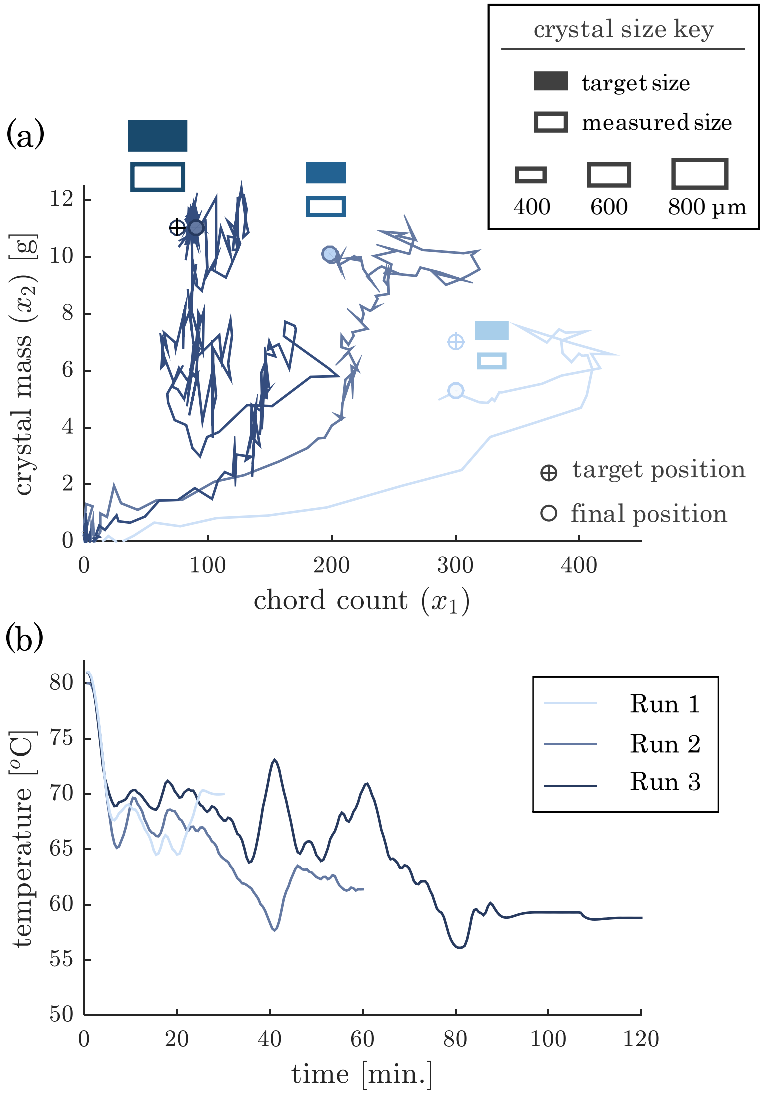

Figure 6.19: (a) Measured trajectories for darapskite salt crystallization under model-based feedback control towards three separate targets. For each target, the measured mass-average crystal size is shown against the target size. (b) Implemented temperature profiles.

6.2.4 Application to paracetamol crystallization

The model-based feedback control strategy was also tested experimentally on paracetamol crystallization. Again, three targets positions and batch times were selected. Table 6.7 summarizes the selected targets and times for testing the optimal model-based control strategy on paracetamol crystallization.

Table 6.7: *Run targets.* Model-based control applied to paracetamol system.

Run	**Target Position (count, mass [g])**	**Target Crystal Size mass-weighted mean [μm]**	**Target Batch Time [min.]**
1	(4000, 20)	223 (82–364)*	150
2	(1000, 20)	354 (146–497)	360
3	(275, 20)	544 (265–692)	360

*95% confidence interval based on mass-per-count–size correlation given in Figure 4.8

Control policies. As was the case for darapskite crystallization, the optimal state-feedback control policies obtained for paracetamol can be shown as time-varying color maps. For example, Figure 6.20 shows representative color maps of the optimal policy for reaching Target 1.

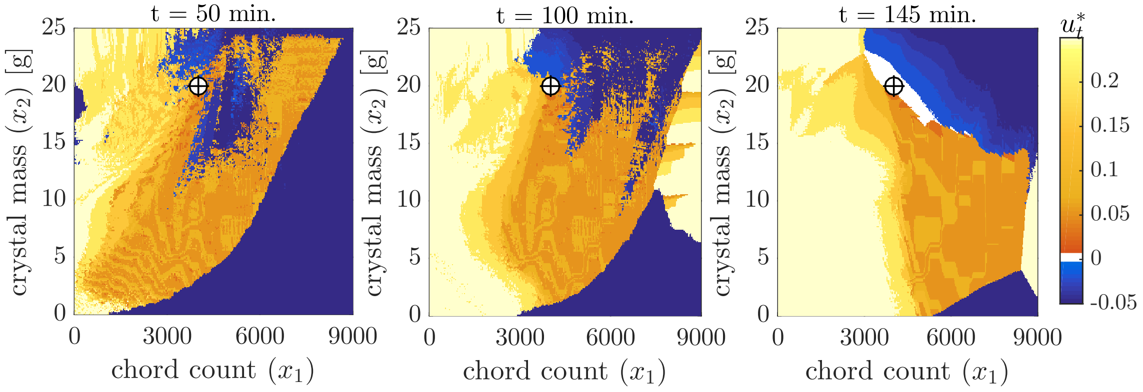

Figure 6.20: Color maps of the optimal state-feedback policy, π, for paracetamol crystallization towards Target 1 at three representative times during the process (during the control run the map updates every 5 minutes).

Here again we see some similarities with the rule-based schemes. Figure 6.21 illustrates this, showing the Zones used for FZ-aDNC overlaying the model-based control policy at 145 minutes.

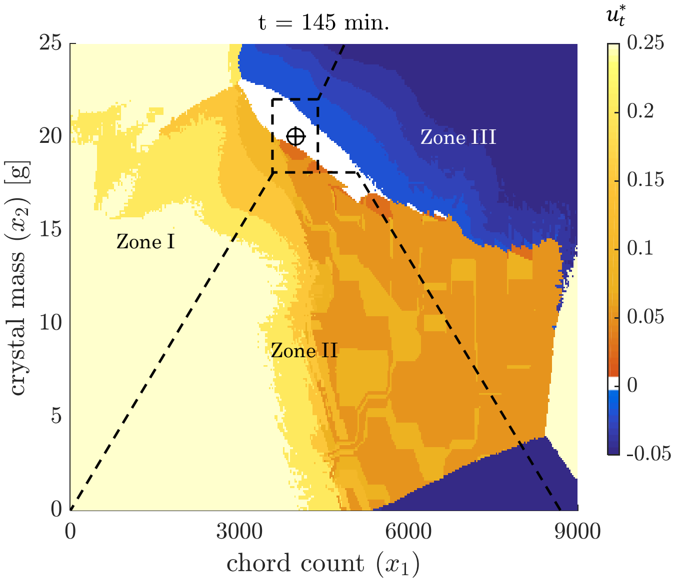

Figure 6.21: FZ-aDNC Zones overlaying the model-based policy, π, for paracetamol crystallization towards Target 1 at 145 minutes.

Although the model-based policy is similar in general concept to the rule-based schemes, the details incorporated through the model again provide an extra level of sophistication. The model-based controller here applies more advanced time-varying control in the *difficult* regions (above and to the left or below and to the right of the target); it also varies the input level with the distance-to-target, slowing the movement down as the target is approached, and it adjusts the effective target zone as the run progresses.

Model-based control of paracetamol requires tuning. In Section 5.2.5 we pointed out that the Markov state model for paracetamol deviates markedly from the observed dynamics in a number of situations. In addition, when comparing the Markov model for paracetamol with that obtained for darapskite (Figure 5.15 compared with Figure 5.14) we noted that we have less influence over the paracetamol crystallization trajectory. This has a consequence: the model-based control for paracetamol is less robust and requires the implementation of faster temeperature maniuplations. And this means we have to *tune* the model-based controller. In particular we have to adjust the internal gains in the PI temperature loop.

The gains must be enough to inact fast temperature changes, but low enough to avoid significant overshoot. Figure 6.22, for example, shows the measured MC trajectory for a paracetamol crystallization under model-based control towards the third target when the internal loop was poorly tuned. In this case, the controller initially applied temperature cycles that caused crystallization followed by complete dissolution. After this initial period with no progress, the controller then applied a complex temperature profile that moved the position towards the target but ultimately missed. In this case, the PI feedback loop used to control temperature was made too aggressive (in an attempt to achieve rapid temperature changes) and this had a side-effect that undermined the control performance: the temperature frequently overshot. A clear example of this is seen at the start of the run given in Figure 6.22 where temperature overshoot inadvertently causes the complete dissolution of crystals, bringing the state back to the origin.

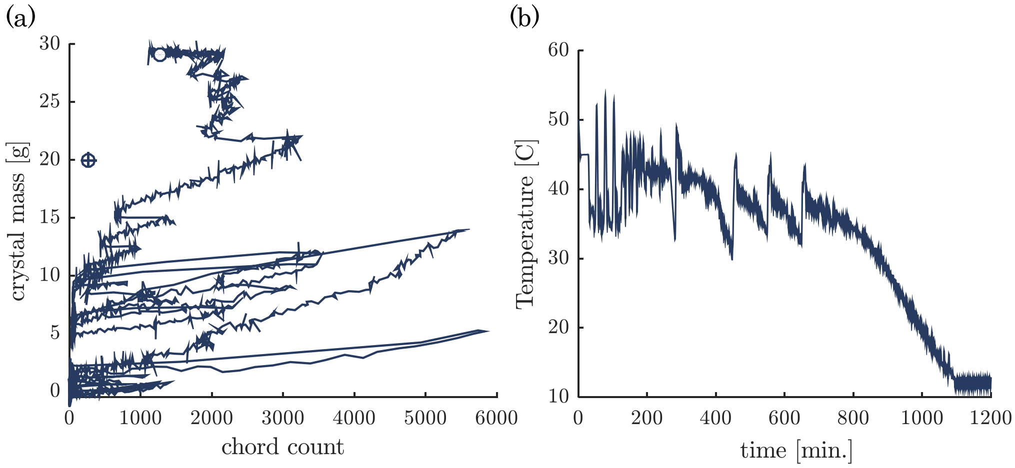

Figure 6.22: *Poorly tuned model-based control over paracetamol:* (a) measured trajectory for paracetamol crystallization under model-based feedback control towards the third target; (b) implemented temperature profile.

Model-based control of paracetamol crystallization. After some adjustments to the internal loop we find that the model-based control strategy can be applied to accurately control the final MC position. Figure 6.23 summarizes experimental results of the model-based control strategy applied to paracetamol crystallization towards the three different targets.

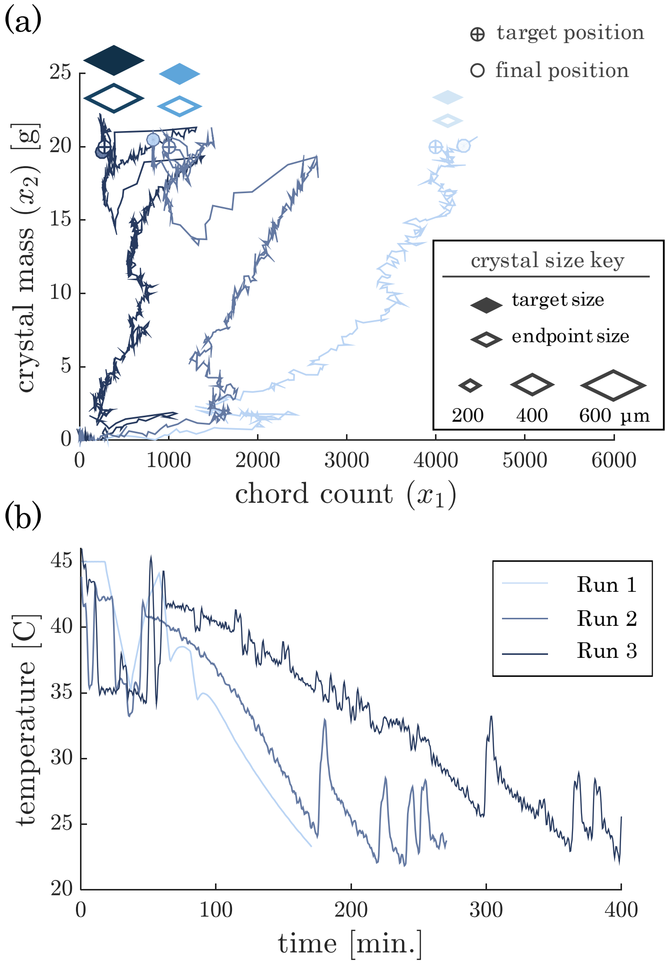

Figure 6.23: (a) Measured trajectories for paracetamol crystallization under model-based feedback control towards three separate targets. For each target, the final inferred crystal size is shown against the target size. (b) Implemented temperature profiles.

In these runs, the optimal state-feedback controller guided the MC trajectory to reach a position near the target position through the application of complex temperature profiles. This presumably results in the production of crystals of near target sizes:

- Run 1 under optimal control produced crystals with an inferred mean size of 225 μm compared to a target of 223 μm;
- Run 2 under optimal control produced crystals with an inferred mean size of 382 μm compared to a target of 353 μm; and,
- Run 3 under optimal control produced crystals with an inferred mean size of 563 μm compared to a target of 544 μm.

(note that *inferred* sizes are listed here. These size are calculated from the final mass-count position using the correlation given in Figure 4.8.)

6.2.5 Summary of model-based control

Model-based control was applied to darapskite salt crystallization to optimally produce crystals of target sizes in the specified batch times. The key result here is the demonstrated MC endpoint control—and thereby size control—shown in Figure 6.19(a), but the input profiles themselves (Figure 6.19 (b)) are also worth remarking on. There is a sharp difference between these temperature profiles and the concave temperature profiles that are usually obtained from optimal population balance model calculations. Rather than simply adjusting the curve of a strictly decreasing temperature profile, more complex profiles that include heating stages are generated. These temperature profiles are the product of a model-based policy that leverages the asymmetry between crystallization and dissolution and is also able to adapt to run-specific dynamics to ultimately achieve better control.

Model-based control was also applied to paracetamol crystallization. Here we find that the strategy is not as robust and the control loop must be tuned or calibrated to achieve reliable control. There are a few reasons why the model-based strategy is more robust when applied darapskite crystallization than when applied to paracetamol crystallization, namely: the MC space Markov model is a more accurate representation for darapskite crystallization than it is for paracetamol crystallization; and, the model for paracetamol crystallization suggests that very rapid temperature manipulations—which are difficult to accurately implement—are required to achieve optimal control over paracetamol crystallization.

6.3 Chapter Conclusions

The results presented in this chapter speak directly to the core utility of the MC framework: the MC framework provides a useful paradigm for establishing feedback control over the mean crystal size. To be specific, we have seen that—within the MC framework—both rule-based and model-based feedback schemes can be developed and these schemes can then be applied in practice using recently developed PAT tools; in applying these schemes to the two experimental systems, *explicit* size control is demonstrated—that is, the ability to pick out a specific target mean size prior to the run and then hit this target. To our knowledge, crystal size control has not previously been demonstrated (experimentally) to the same extent for unseeded batch cooling crystallization.

A Framework for Understanding and Controlling Batch Cooling Crystallization

Conclusions and Future Work

CHAPTER 7

THESIS SUMMARY

THIS thesis develops a framework for addressing a specific process engineering control task: control over the mean size of crystals produced by batch cooling crystallization.

Depending on your perspective, this can seem like a simple task to address. On one hand, we have a single input variable to adjust—temperature—in order to control a single output variable—the mean crystal size. Closer inspection, however, reveals complexity. The operation is dynamic: the temperature change over time drives actions at the molecular level—crystal nucleation, growth, dissolution, agglomeration and breakage—and these underlying processes collectively dictate the number and size of the crystals produced. This makes for a convoluted input-*profile*–output relationship that cannot be easily revealed with simple, trial-and-error-type experiments. Instead, a framework is needed to understand the crystallization dynamics and establish the relationship between the input temperature profile, these dynamics, and the crystal size.

One way to build this framework is to start from the bottom up: modeling the evolution of the crystal population according to the underlying processes by which individual crystals form and evolve. This line of development has led to the well-established population balance (PB) framework, which is both conceptually stimulating and mathematically complete. But it also has drawbacks for application—in particular, it is not conducive to *feedback* control.

My intent in this thesis has been to show that an alternative perspective is useful. Rather than attempting to model crystallization dynamics at the molecular or single-crystal level and then build up, the collective dynamics can be viewed more simply as the evolution of two key aggregate properties that can be *measured*: the crystal mass and count. Taking this perspective reveals a framework—labeled the mass-count

(MC) framework—that is well-suited to address the crystal size control problem.

Using the MC framework, crystallization dynamics are understood intuitively as movement in a 2D space. What is more, the perspective shifts the conceptual understanding of the control problem: we now see the crystal size control problem as a trajectory endpoint control problem. And this clarifies the route to achieving crystal size control. To achieve size control we need to: (1) learn the dynamics or how to move in MC space, and then (2) develop schemes for driving to a target positions in the space—thereby controlling the mean crystal size. Because we can monitor the crystal mass and a measure related to the crystal count, both tasks can be readily achieved in practice. In particular, we are able to apply data-driven strategies to learn the dynamics and then leverage real-time monitoring and the framework interface to develop effective *feedback* control schemes.

The applicability of the MC framework to establish control by this route was demonstrated on widely different experimental systems: darapskite salt crystallization from water and paracetamol crystallization from ethanol. For both systems, we learned to move in MC space—the first step—using two data-driven strategies: observational learning, in which a conceptual understanding of the movement was revealed through visual observations of collected run data; and machine learning, in which a low-dimensional mathematical model of the movement was identified from collected run data. In parallel with the methods used to learn the dynamics, two types of feedback control schemes were then developed: rule-based feedback schemes, which utilized the conceptual understanding of the dynamics; and model-based feedback schemes, which utilized the mathematical model of the dynamics developed from machine learning. Using these schemes, *explicit crystal size control was demonstrated* for both experimental system. These control results provide the strongest support for the utility of MC framework.

CHAPTER 8

FUTURE WORK

MY intention in sitting down to write this thesis was to provide a complete story around the mass-count framework. There are, nevertheless, questions that remain unanswered and topics relating the framework that I would have liked to explore further. Listed here are a few lines of inquiry that I think could be particularly useful for improving our understanding of crystallization dynamics, refining the MC framework, and improving crystal size control.

8.1 Exploring the Relationship Between the Frameworks: PBMs from the MC Perspective.

The PB and MC frameworks give different perspectives of the same dynamic process. I would have liked to have spent more time reconciling these perspectives, as I expect that work along these lines could reveal an improved understanding of crystallization dynamics.

One way we might explore the relationship is by analyzing the mass-count (MC) trajectories predicted by population balance (PB) models. This exploration could help us understand how to interpret MC space movement from a theoretical or mechanistic vantage point—that is, this exploration could help us understanding what underlying crystallization/dissolution mechanisms are responsible for certain MC space movements. In addition, this line of inquiry might answer questions relating to PB models and the MC framework that have repeatedly come up: can PB models be translated to MC-movement models? If so, do such models line up with the empirical Markov state models we have learned from data? And can the PB models then be used to establish model-based feedback control?

Preliminary (and incomplete) results showing the MC trajectories predicted by PB models.

Trajectories resulting from a crystallization-only PB model (reduced to moments)

Early in my PhD, I constructed population balance models of the crystallization dynamics for different salts and reduced these using the method of moments. For sodium nitrate crystallization from water, I used the following expressions for nucleation and growth:

$$J = J_{0,1}\, exp\left(-E_J/(T(T-T_{\mathrm{S}}))^2\right) + J_{0,2}\sigma^b\mu_2; \text{ and}$$

$$G = k_g\, exp\left(-E_g/T\right)\sigma^g;$$

where μ_2 is the second moment of the crystal size distribution and the parameters $J_{0,1}$, $-E_J$, $J_{0,2}$, b, k_g, E_g, and g were found fitting the model to concentration data.

We can explore this model from the MC perspective and try to learn what it says about the dynamics. For example, Figure 8.1 shows the MC trajectories predicted by the PB model for two linear cooling crystallizations—one under moderate cooling and one under slow cooling.

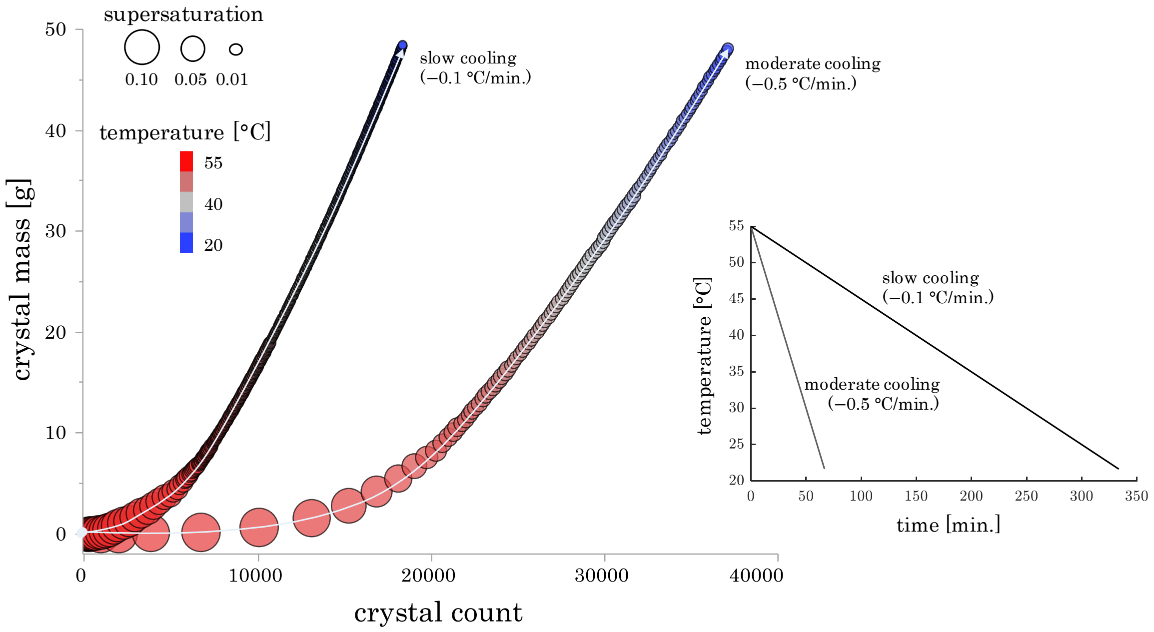

Figure 8.1: Mass-count trajectories predicted by a PB model (reduced to moments) for sodium nitrate crystallization under slow linear cooling and moderate linear cooling.

Here we see that the simple model constructed using the PB framework clearly captures an empirical observation: slower cooling results in a steeper MC trajectory and produces larger crystals. Further examination along these lines might reveal how we can use the shape of the trajectory in MC space to make inferences about the underlying nucleation, secondary nucleation and growth mechanisms.

Trajectories resulting from a PB model that includes both crystallization and dissolution

A recent paper by Yang and Nagy [Yang and Nagy, 2015] puts forth a population balance model (reduced to moments) that has been adapted from [Lindenberg et al., 2009] to include dissolution:

Nucleation, Growth and Dissolution [Yang and Nagy, 2015]

$$S = \frac{C}{C^*} \tag{8.1a}$$

$$\underbrace{G = k_{G1}\, exp\,(-k_{G2}/RT)\,(C^*(S-1))^{k_{G3}}}_{\text{growth rate model}},\ \text{if } S \geq 1 \tag{8.1b}$$

$$\underbrace{B = k_{B1}\, exp(-k_{B2}/RT)\, exp(-k_{B3}/ln^2 S)}_{\text{nucleation rate model}},\ \text{if } S \geq 1 \tag{8.1c}$$

$$\underbrace{D_G = -k_{G1}\, exp\,(-k_{G2}/RT)\,(C^*(S-1))^{k_{G3}}}_{\text{dissolution rate model}},\ \text{if } S < 1 \tag{8.1d}$$

$$\underbrace{B = -k_{B1}\, exp(-k_{B2}/RT)\, exp(-k_{B3}/ln^2(1/S))}_{\text{extinction rate model}},\ \text{if } S < 1 \tag{8.1e}$$

With this model, we can examine the MC trajectories predicted for different cooling-*heating* operations. Two different simulated crystallization-dissolution runs are given here.

The first is shown in Figure 8.2. In this simulated run, slow cooling is applied initially, causing crystallization. This is then followed by slow heating to completely dissolve the formed crystals. Here we see that the *crystallization* trajectory qualitatively aligns with what we have observed in experiments: it moves from the origin off to the right and then upwards are the supersaturation is depleted. But the *dissolution* trajectory is markedly different from what we observe in experiments and also different from what we would expect, physically: the crystal mass is reduced to zero while the count is nearly unchanged and remains high at the end of the run after full dissolution.

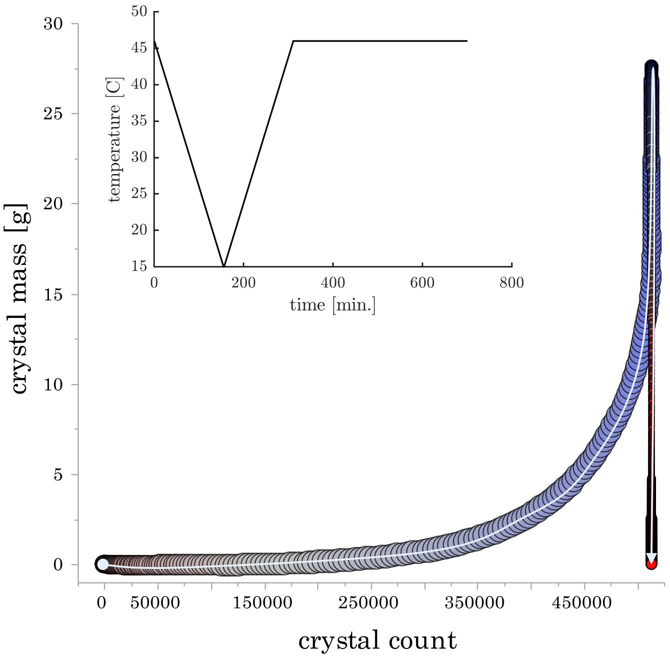

Figure 8.2: Mass-count trajectory predicted by the moments model given by [Yang and Nagy, 2015] for paracetamol crystallization when slow linear cooling is followed by slow heating. *Inset:* applied temperature profile.

A second crystallization-dissolution simulation reveals further odd model-predicted dynamics. Figure 8.3 shows the MC trajectory predicted by the model under faster cooling followed by faster heating. Under cooling, the crystallization trajectory roughly aligns with what we observe in experiments (although the counts are unexpectedly lower than what we observed for slower cooling in the previous simulation). But, here again, the dissolution trajectory deviates from experimental observations and physical expectations: the crystal count, in this case, is completely depleted while the crystal mass remains high.

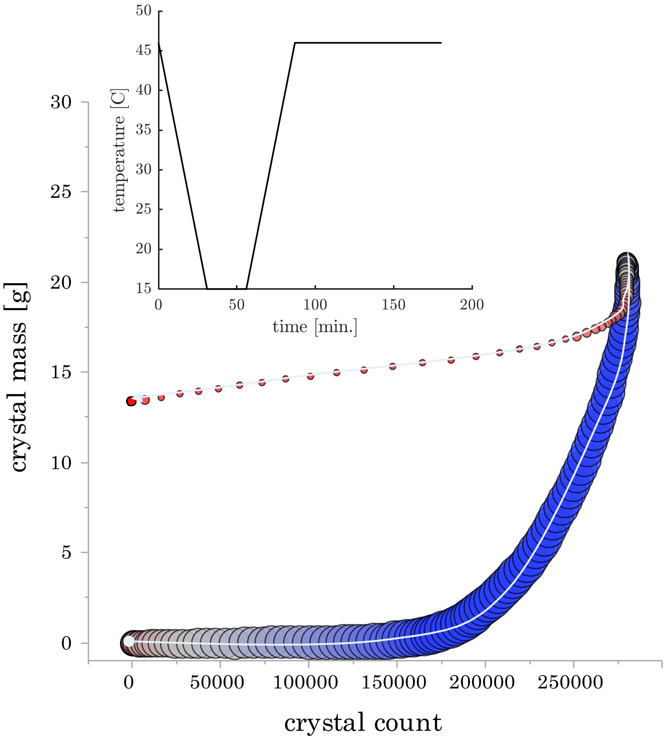

Figure 8.3: Mass-count trajectory predicted by the moments model given by [Yang and Nagy, 2015] for paracetamol crystallization when fast linear cooling is followed by fast heating. *Inset:* applied temperature profile.

The MC trajectories shown here highlight a shortcoming of this PB model: the current crystal state is not explicitly included in the rate expressions. (And, a brief survey of other population balance models suggests that this is not unique to this particular model). In experiments we observe that the current crystal state *does* have a significant impact on the dynamics—especially the dissolution dynamics. This is seen in the empirical Markov models we learn from data: Figures 5.14 and 5.15 indicate that crystal state (the mass and count) influence the forward movement.

Further work along these lines may reveal how to incorporate crystal state dependence into rate expression for the underlying nucleation, growth and dissolution. In addition, it may reveal what sort of underlying mechanisms are required to produce the type of complex trajectories we observe in experiments when more complicated temperature profiles are implemented (for example, Figures 5.17 and 5.19).

8.2 Refining MC Control Schemes

The feedback control schemes presented in this thesis pass a practical test: they *worked* when experimentally applied on a few test cases. But, additional examination is needed to fully characterize the schemes and understand their limitations. In addition, there is significant room for tuning, adjusting, and reworking the schemes presented here in order to improve control performance.

Understanding the reliability of control: replicate runs. The feedback control algorithms developed here *appear* to be reliable (perhaps, with the exception of the model-based scheme applied to paracetamol crystallization). But how reliable the schemes are, and how much the results can be expected to deviate from run to run remains an open question.

There are many ways this could be approached. One practical way is through replicate runs—that is, applying the same scheme to control to the same target multiple time. Replicate runs of this nature were applied with the sGATEC scheme towards Target 2, as shown in Figure 8.4.

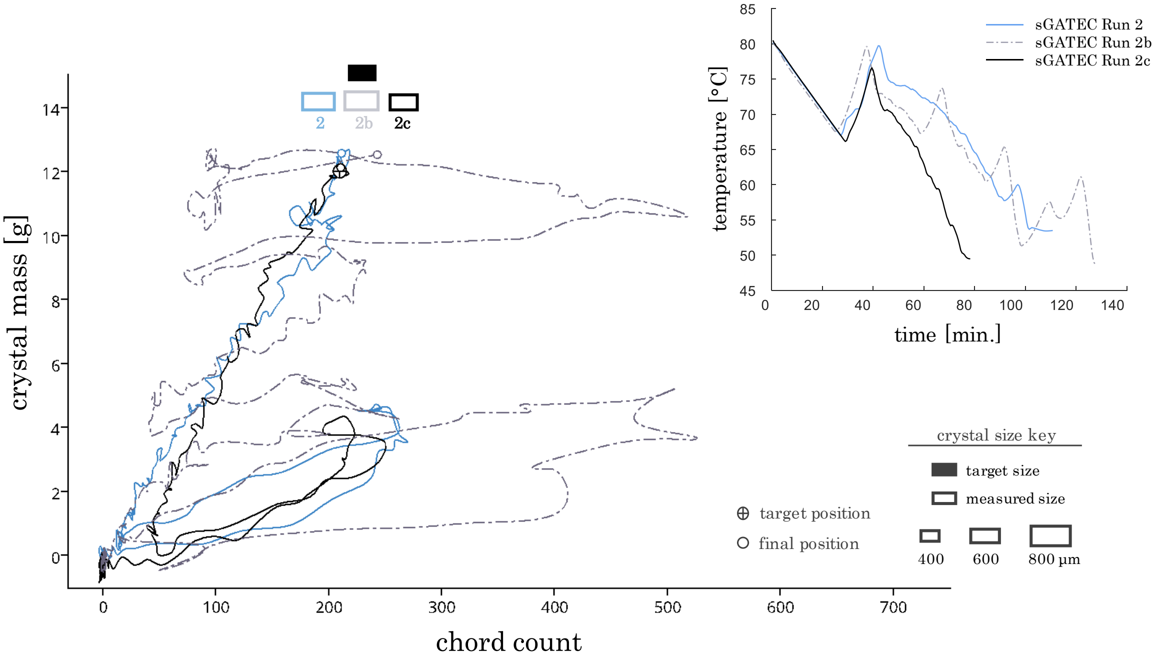

Figure 8.4: Replicate sGATEC runs towards Target 2

In these replicate runs we see that the controller is able to guide the trajectory to reach a position close to the target in each case and thereby produce crystals with measured mean sizes close to the target size. But the path to the endpoint varies from run to run—illustrating process variability and demonstrating why feedback is essential.

To more definitively make claims on the reliability of the feedback schemes and characterize the run-to-run variability, more extensive replicate studies should be implemented.

Tuning the control loop and treating temperature as the input. The controllers developed here often require specification of a number of control parameters. In addition, each feedback control scheme developed in this thesis is implemented with a cascade feedback loop, in which proportional-integral (PI) feedback control over the temperature is implemented internally. I have not rigorously addressed how the control parameters or the PI coefficients should be selected. Developing a tuning method here could improve the control. This is clearly seen with the model-based controller applied to paracetamol crystallization. In addition, further refinement on the model-based control could be achieved by treating the temperature as the true input and explicitly incorporating the input dynamics in the optimal control calculations.

Alternative rule-based control schemes. Within the MC framework, it is easy to dream up different rule-based control algorithms (and I have found this to be one of the more enjoyable tasks in my PhD work). Those presented in Chapter 6 are only a very small subset of the schemes that can likely be implemented.

Especially worth pursuing, in my opinion, are new rule-based schemes that incorporate knowledge of the recent movement. The recent movement in a run can be very useful information in predicting the future movement but has not been incorporated in the presented strategies.

Alternative model structures and improved model-based control. We have used a simple mathematical Markov model to capture the crystallization and dissolution dynamics in terms of MC movement. To be sure, there are advantages of this type of model in performing computation mathematics. But it also is limited in that must predict the dynamics from only the *current* state and input. This was seen to be a shortcoming in modeling paracetamol crystallization and dissolution dynamics. Other mathematical model structures that incorporate past history in predicting the dynamics may show improved predictive capabilities. This, in turn, could lead to improved model-based control.

8.3 Extending the MC Framework

The mass-count framework has received positive feedback from colleagues in academia and industry. But the framework is also somewhat limited in its current application: control of the *mean* size of crystals produced by *unseeded batch cooling* crystallization. I see potential to extend the MC framework and broaden the application space.

8.3.1 Controlling more than just the *mean* crystal size

It may be possible to use the MC framework, or a slight variant, to control more than just the *mean* crystal size.

Controlling the crystal size *distribution* by MC trajectory *path* control. The final MC position specifies the mean crystal size. It does not, however, specify the crystal size *distribution*. Figure 8.5 illustrates this: here we see three distributions that are clearly different despite having the same mean size.

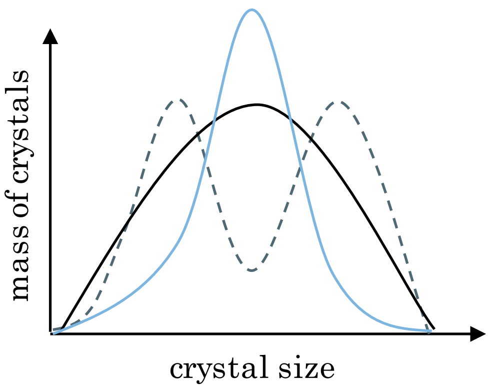

Figure 8.5: Varying crystal size distributions with the same mean.

In some cases, just controlling the mean crystal size may not be the full objective. It might, instead, be important to control the size distribution. The question is: can the MC framework be used to establish control over more detailed properties of the size distribution?

I think the answer is yes. But it requires more than endpoint control. To control the size distribution, the MC trajectory *path* must be carefully controlled. For example, the crystals size distributions shown in Figure 8.5 may be achieved by controlling the MC trajectories as shown in Figure 8.6.

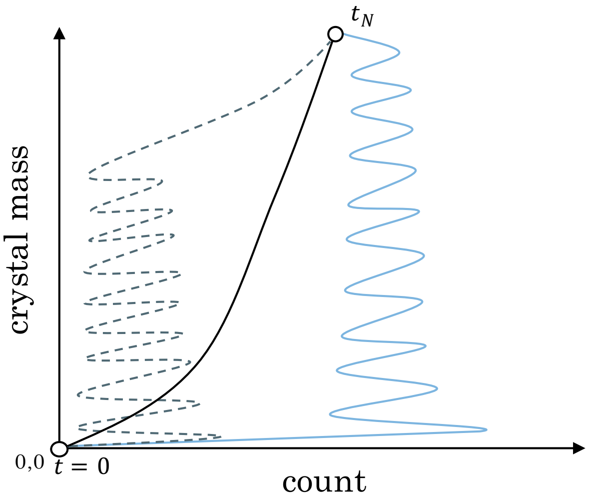

Figure 8.6: Hypothetical MC paths that might be used to produce the size distributions given in Figure 8.5.

Although this is hypothetical and should be experimentally confirmed, the idea seems sound. A bi-modal distribution can be produced by first generating a population of large crystals and then generating a number of small crystals on top of those. This can potentially be achieved with the path traced out by the dashed line in Figure 8.6.

Conversely, a narrow, mono-modal size distribution can be achieved by rapidly producing crystal nuclei at the same time, then using temperature cycles to ratchet the size of those crystals upwards while periodically dissolving any new fines. The trajectory given by the light blue line should accomplish this.

Finally, a broad distribution can be achieved by slowly generated new crystals while existing crystals grow. The path traced out by the solid black line is expected to accomplish this.

Crystal state expansion: creating an MC+ framework. If additional relevant crystal state properties could be measured online, the state representation could be expanded and along with it the scope of the control. For example, if it were possible to accurately measure the average crystal purity online, this could be treated as a third state variable. The expanded (MC+) framework, would then view crystallization dynamics through the evolution of the mass, count, and average crystal purity. Given the right training data, it is possible that the same type of data-driven modeling and dynamic programming strategy could be used to identify operating policies for controlling the mean crystal size and purity.

While I expect this line of exploration would be fruitful—provided the online measurements become available—there are certainly drawback to expanding the crystal state representation. This is, after all, a key difference between the population balance framework, which I claimed was hard to use, and the mass-count framework, which I have claimed is conducive to application. Not least among the drawbacks: if the crystal state is expanded to be greater than three-dimensional, we loose the ability to easily visualize the dynamics. In addition to this, model-based computation becomes more difficult and control over the full state will likely be diminished.

8.3.2 Extending the MC framework to other crystallization operations

It is possible that the MC framework can be extended to operations other than unseeded batch cooling crystallization. The biggest challenge in extending the MC framework to these operations is in closing the mass balance so that the total crystal mass may be monitoring in real time via solution concentration measurements. Given a direct measure of the total crystal mass, this challenge can be largely avoided and the extensions made more easily.

Seeded batch cooling crystallization. The easiest extension of the MC framework—as presented in this thesis—is to *seeded* batch cooling crystallization. Seeding simply jumps the MC position away from the origin. The mass jump must be incorporated through knowledge of the mass of seeds added. The count jump can be monitored directly.

The MC framework can then be used in much the same way as it is for unseeded batch cooling crystallization, with one exception: it is likely that the dynamics—as seen by MC movement—in the instants immediately following seed addition will be different from the dynamics at other points during the run. Revealing these post-seed dynamics is important for crystal size control. It can also answer interesting question related to processing the seeds. For example, how should the seeds be processed in order to promote growth over secondary nucleation?

Continuous cooling crystallization. After seeded batch cooling crystallization, the next easiest extension of the MC framework is to continuous cooling crystallization. To close the mass balance and extend the MC framework to continuous cooling crystallization, streams into and out of the crystallizer must be characterized. Including the flow rates, composition in, and solid fraction out. The flow rates are typically measured and the composition in is usually known. The solid fraction out may be inferred through well-mixed assumptions.

Provided the framework can be extended, the framework and feedback interface might be used to maintain the MC position at a fixed location—adapting to disturbances and producing a consisting crystal product.

Batch antisolvent crystallization. The MC framework can also conceivably be extended to antisolvent crystallization. In this case, the input in no longer the temperature profile, but rather the anti-solvent/solvent addition profile. Again, to extend the MC framework, the mass balance must be closed. Also, solute concentration monitoring must be accurate at varying levels of solvent/antisolvent.

Here again there is an interesting potential use of the MC framework: we might identify feedback control policies that indicate the appropriate anti-solvent/solvent additions over time in order to control the dynamics and produce crystals of the target size.

8.3.3 Beyond crystallization

The modeling and control strategy engendered by MC framework model-based approach to controlling crystallization—state reduction followed by data-driven modeling and dynamic programming—has utility beyond crystallization. Indeed, this is a strategy that has been frequently applied in the Grover Group to a variety of complex systems with good practical success [Xue et al., 2014, Griffin et al., 2016b]. The rigorous development of this approach as a general methodology is a worthwhile pursuit.

APPENDIX A

MONITORING MULTICOMPONENT ELECTROLYTIC SOLUTIONS

Monitoring the solution composition and supersaturation is crucially important to understanding and controlling batch cooling crystallization. This chapter outlines the contributions we made in [Griffin et al., 2014] and [Griffin et al., 2015d] that enable multicomponent solution monitoring.

Composition monitoring. The solution composition can potentially be monitored online with ATR-FTIR, as outlined in §4.3. But this first requires the development of a calibration model. For single-component crystallization systems this can usually be accomplished with well-established chemometric techniques, such a PLSR, that are now widely available to be applied in an off-the-shelf manner. However, tailored chemometric strategies may be required for multicomponent crystallization systems—like darapskite salt crystallization from a multicomponent electrolytic solution. This section describes a tailored regression strategy, termed robust parameter support vector regression (RPSVR), that is useful for establishing accurate multicomponent concentration monitoring [Griffin et al., 2014].

Supersaturation monitoring. The supersaturation for single component, non-dissociating solutes is given by the simple expression

$$\sigma \equiv \frac{C - C^*(T)}{C^*(T)},$$

where C is the current concentration of the solute in solution and $C^*(T)$ is the solubility concentration at the given temperature. This measure can be monitored online using ATR-FTIR and temperature measurements, provided that we have an IR-to-concentration calibration model and the solubility-temperature relationship is known.

It is a little trickier to define supersaturation for multicomponent solute that dissociates in solution. Obtaining the solubility-temperature relationship is also more difficult. This chapter gives the molar supersaturation—a measure of supersaturation for dissociating salts—and also gives a method for quickly obtaining the solubility of multicomponent solution, termed the solubility trace method [Griffin et al., 2015d].

A.1 Robust Parameter Support Vector Regression (RPSVR)

In experimental System I—darapskite salt ($Na_3SO_4NO_3{\cdot}H_2O$) crystallization from water—we need to simultaneously monitor the concentration of the anions SO_4^{2-} and NO_3^-. More generally, we have been interested in monitoring complex electrolytic nuclear waste solutions that contain many different anions, including those in darapskite and also commonly nitrite, NO_2^-, and carbonate, CO_3^{2-}.

In building calibration models for inferring the concentration of each of these anions in a multicomponent solution, we run into a challenge: the characteristic infrared absorbance peaks for the different anions overlap. As a result, we are forced to use minor spectral features to extract the information from a convoluted solution spectrum, and this can be problematic when changing process conditions introduce error in the measurement.

To achieve accurate concentration monitoring in this situation, we needed some way of expertly separating the signal from the noise; Robust Parameter Support Vector Regression (RPSVR) was the regression algorithm we came up with for doing this [Griffin et al., 2014].

A.1.1 IR-to-concentration calibration model: notation

The goal of calibration, here, is to identify a function that accurately infers the concentration of a given component in solution from the infrared absorbance spectrum measured for that solution. To describe this function, and the process of learning it from data, some notation needs to be introduced.

Input: features of the solution infrared absorbance spectrum. Figure A.1 shows an infrared absorbance spectrum measured by ATR-FTIR for a multicomponent salt solution. The y-axis shows the absorbance (a measure of the amount of light absorbed by the solution) for light of different frequencies, indicated by the x-axis.

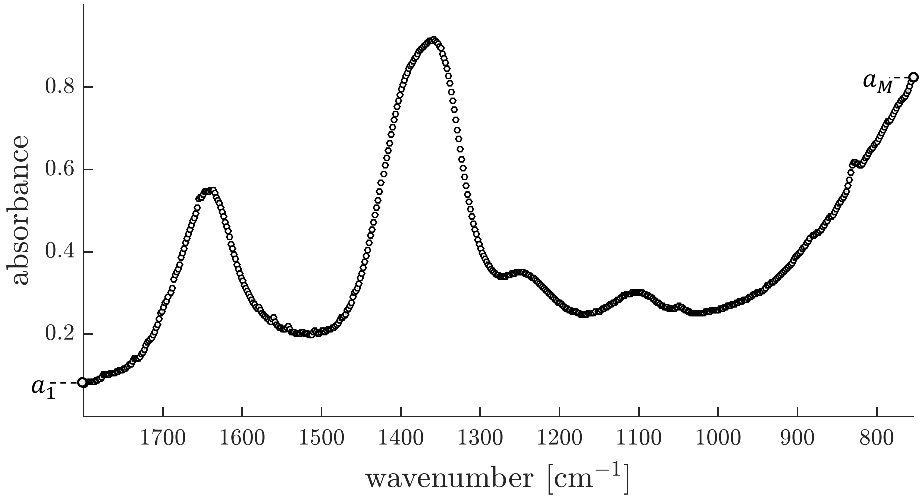

Figure A.1: Infrared absorbance spectrum.

This information is conveniently represented as a vector

$$\mathbf{a} \equiv \begin{bmatrix} a_1 \\ \vdots \\ a_M \end{bmatrix}, \tag{A.1}$$

where each element contains the measured absorbance for a particular frequency of light.

The input to a calibration model are *features* of the spectrum, denoted by $\mathbf{x}$. A feature, for example, might be the height of a select peak. Each feature may also simply be a single element of the infrared absorbance spectrum vector, $\mathbf{a}$. In any case, the features must be a function of spectrum vector:

$$\mathbf{x} = \phi(\mathbf{a}). \tag{A.2}$$

Output: concentration of a solution component. The output of the calibration model—what the model is trying to predict—is the concentration of a single component in solution:

$$y = C.$$

For multicomponent solutions, multiple calibration models are needed—one for each solution component of interest.

Calibration model. A calibration model *infers* or *predicts* the concentration from the input features of the infrared absorbance spectrum. To distinguish this inference from the true concentration, it is denoted with a hat:

$$\hat{y} = f(\mathbf{x}),$$

where $\hat{y}$ is the concentration *predicted* by the calibration model, f, for the input features $\mathbf{x}$. In this work, and for most IR-to-concentration calibration models, the function f (the calibration model) is chosen to be affine—that is, a linear function of $\mathbf{x}$ with an added constant. This type of function can be written succinctly:

$$\hat{y} = f(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x} + b, \tag{A.3}$$

where $\mathbf{w}$ and b are parameters that must be learned through calibration.

A.1.2 Calibration steps: training data collection, feature selection, and regression

Calibration (i.e. the building of a calibration model) consists of three steps: (1) training data collection, (2) input feature selection, and (3) regression. Each step is important for building an accurate calibration model.

Training data. Training data consists of a series of spectrum-concentration pairs:

$$\{(\mathbf{a}^{[1]}, y^{[1]}), \cdots, (\mathbf{a}^{[N]}, y^{[N]})\}. \tag{A.4}$$

This data should be carefully collected to reveal the connection between the concentration of the component of interest, y, and the infrared absorbance spectrum, $\mathbf{a}$. As such, y should be varied over the full experimental range and should be changed independent of any disturbance variables—e.g. temperature. In addition, if the solution has multiple components, the concentration of each should be varied independently.

Input feature selection. The task of selecting input features in not a simple one. Nor one that can be made independently of the training data available and the regression algorithm to be used. It often boils down to a judgment call, with the guiding principles that the selected features should be minimal, clear and unaffected by disturbance variables. For single component solutions it is common to select a characteristic peak and use the height or area of the peak as the only input feature. For multicomponent solutions it is more common to use data-informed projection methods, like principle component analysis, to automatically select the input features or use the full spectrum as the input. In any case, once the features have been decided on, the training data is converted to input–output ($\mathbf{x}$–y) data.

Regression. The final step of calibration is regression. This is the process by which the unknown calibration model is fit to the input–output training data. Regression is an optimization problem. For an affine calibration model, as given in (A.3), the optimization problem is written as:

$$\begin{aligned} \underset{\mathbf{x},\, b}{\text{minimize}} \quad & \Phi\left(\hat{y}^{[1],\ldots,[N]}, y^{[1],\ldots,[N]}\right) \\ \text{subject to} \quad & \hat{y}^{[i]} = \mathbf{w}^{\mathrm{T}}\mathbf{x}^{[i]} + b, \ \text{for } i = 1, \ldots, N; \end{aligned} \tag{A.5}$$

where Φ is the objective function to be minimized.

A common objective function is the *sum-squared-error*:

$$\sum_{i=1}^{i=N} \left(\hat{y}^{[i]} - y^{[i]}\right)^2 .$$

In which case the regression is termed least-squares regression.

But this is not the only objective function that can be used, and, as we will see with the development of Robust Parameter Support Vector Regression (RPSVR), the choice of objective function can have a significant impact on the general accuracy of the calibration model learned.

A.1.3 The goal is to find a model with *general* accuracy, not necessarily a tight fit to the training data.

On first pass, it might seem like we are trying to find a model that fits the training data as closely as possible. Although this is roughly what is expressed by the least-squares optimization, it is not the *true* objective of calibration. The objective of calibration is to find a model that is accurate *in general.* That is, a model that accurately predicts the concentration from arbitrary spectra measured in the future.

There is a minor distinction here. A model that fits the training data well, may not be accurate in general. In other words, we can *overfit* the training data.

Overfitting. Overfitting occurs because there is error in the training data that convolutes the true underlying relationship between $\mathbf{x}$ and y. If we are not careful, and fit the training data very closely, our model will read into the idiosyncrasies of the data (due to measurement error) and mistake noise for signal.

Parsing the signal from the noise. Overfitting is avoided when the calibration method accurately separates the signal in the training data from the noise. This can potentially be accomplished changing any one of the three calibration steps: acquiring more data at repeat and different points may reveal what underlying relationship persists and what must have been measurement error; a different set of input features may be less corrupted by noise and more parsimonious; the regression algorithm objective function may be altered to yield a model that does not fit as tightly, but generalizes and produces accurate predictions on unseen data. In developing RPSVR, however, we have focused on avoiding overfitting by altering the *regression algorithm*.

A.1.4 Specific challenge: similar solution components and disturbance variables

We aim to monitor the concentration of multiple similar anions, CO_3, NO_2^-, SO_4^{2-} and NO_3^-, in complex electrolytic solutions. As shown in Figure A.2(a), the infrared absorbance signals for these anions overlap and this leads to a convoluted solution spectrum.

Working with a convoluted solution spectrum can make it more challenging to separate the signal from the noise and avoid overfitting. In particular, we are forced to rely on minor spectral features, like those shown in Figure A.2(b), to deconvolute the spectrum. This is not such an issue as long as the error in the measurement is minor and random so that it is uncorrelated with changes in composition in the training data set. Here is where we run into a problem: for our instrument, in addition to random error in the absorbance spectrum, there are also disturbance variables that introduce significant *structured* variance into the absorbance measurements.

As illustrated in Figure A.3 the infrared absorbance spectrum is significantly influenced by at least two primary disturbance variables: (a) temperatures and (b) probe alignments. It is also influenced in a minor way by two secondary disturbance variables: (c) background spectra and (d) solution mixing intensity.

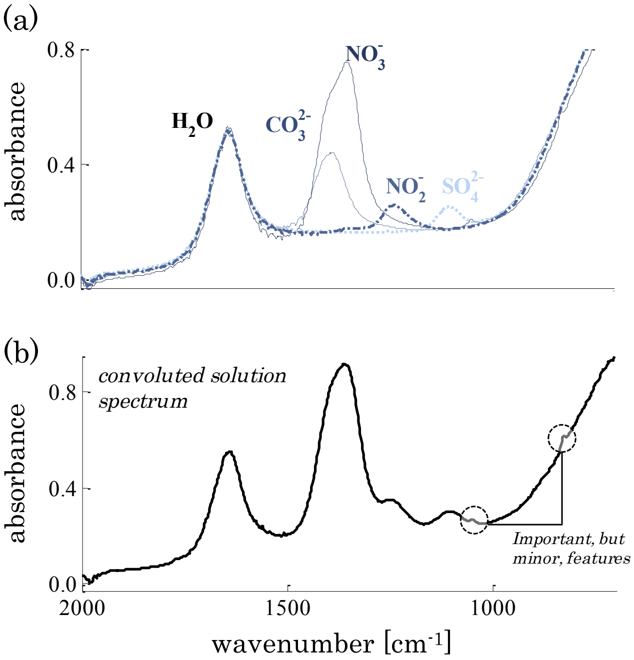

Figure A.2: (a) Infrared absorbance spectra for in the individual components. (b) Convoluted multicomponent solution spectrum.

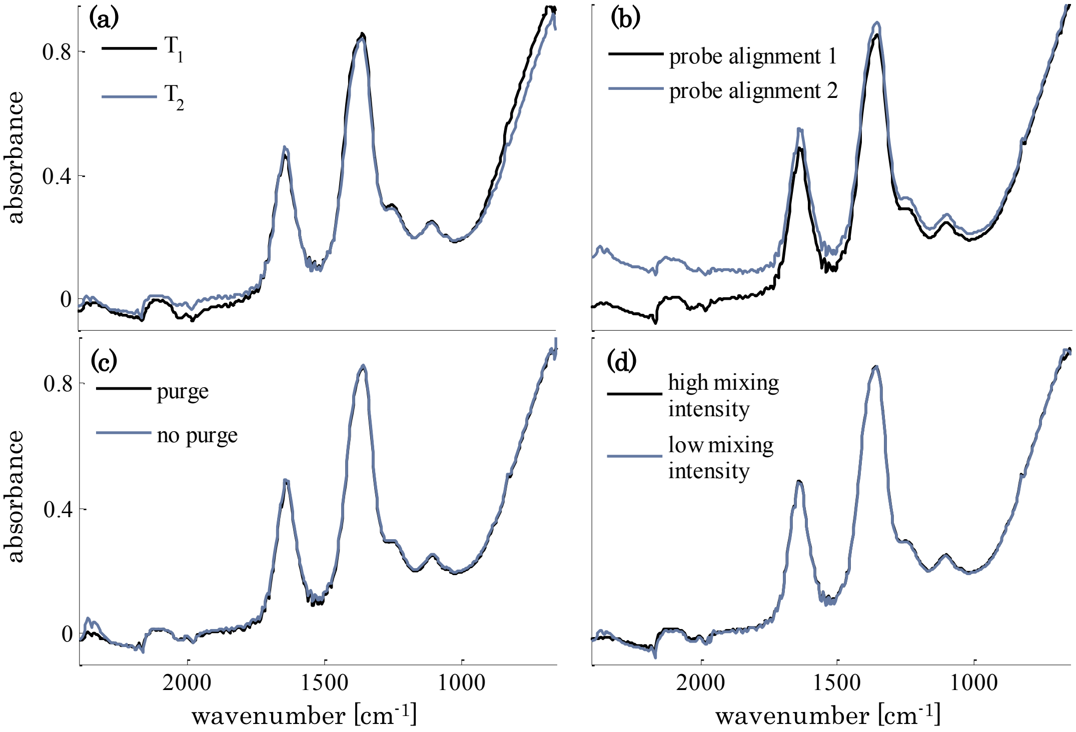

Figure A.3: Measured examples of structured variance in the infrared absorbance spectrum introduced by four disturbance variables: (a) temperature, (b) probe alignment, (c) purge, and (d) mixing intensity.

Having structured variance, or error due to disturbance variables, is especially disruptive to identifying an accurate calibration model. Here is why: some of the disturbance variables cannot be controlled easily (exact probe alignment, for example) and others must be changed throughout the experiments (temperature); if the disturbance variables change while collecting the training data and these changes happen to align with changes in composition, the signal for concentration changes will be convoluted with the variance in the measurement due to disturbance variable changes. In other words, the signal will be tangled with the disturbance-induced noise in the training data.

A.1.5 Changing the regression algorithm to better separate the signal from the noise

As already mentioned, each part of the calibration strategy—data collection, feature selection, and regression—is important for producing an effective calibration model and, moreover, these steps operate together, not independently. Nevertheless, we will single out the regression step as a key to addressing the multicomponent calibration task and work off a large fixed training set of 810 spectra–concentration pairs, with the input feature vector taken to be subset of the infrared absorbance spectrum for frequencies between 764 cm^{-1} and 1507 cm^{-1} (i.e. $\mathbf{x} \subset \mathbf{a}$).

The least-squares regression algorithm. The objective of the least-squares regression algorithm

$$\begin{array}{ll} \underset{\mathbf{x},\, b}{\text{minimize}} & \sum_{i=1}^{N} \left(\hat{y}^{[i]} - y^{[i]}\right)^2 \\ \text{subject to} & \hat{y}^{[i]} = \mathbf{w}^{\mathrm{T}} \mathbf{x}^{[i]} + b, \ \text{for } i = 1, \ldots, N \end{array} \tag{A.6}$$

is to find the calibration model that best *fits* the training data. If the training data is error free or has only normally distributed error in the concentration, this would align exactly with our true objective of finding a calibration model that is *generally* accurate.

However, as we have noted, there is error in the spectral data. And, because this error is structured and on the same order of magnitude as the signal, least-squares regression is susceptible to overfitting. To make the regression algorithm less susceptible to overfitting, we can adjust the objective function.

Adding a penalty term to avoid overfitting. We can add a *penalty* term to the objective function to mitigate overfitting. This penalty term is a function of the model itself and not the data. For a affine calibration model, the simplest penalty term is the following:

$$p(\mathbf{w}, b) = \mathbf{w}^{\mathrm{T}}\mathbb{I}\mathbf{w} + b^2, \tag{A.7}$$

where $\mathbb{I}$ is the identity matrix.

Adding this penalty to the least-squares fit function we get a multi-objective function and a regression algorithm termed ridge regression:

$$\begin{array}{ll} \underset{\mathbf{x},\, b}{\text{minimize}} & \sum_{i=1}^{N} \left(\hat{y}^{[i]} - y^{[i]}\right)^2 + \sigma(\mathbf{w}^{\mathrm{T}}\mathbb{I}\mathbf{w} + b^2) \\ \text{subject to} & \hat{y}^{[i]} = \mathbf{w}^{\mathrm{T}}\mathbf{x}^{[i]} + b, \text{ for } i = 1, \ldots, N, \end{array} \tag{A.8}$$

where σ is an adjustable optimization parameter.

The regression algorithm now reads, "find a calibration model that fits the data, while also keeping the model parameters small." Keeping the parameters small seems arbitrary at first. But it actually serves an important purpose: the smaller the parameters in the affine model, the more insensitive the model is to changes in the input feature vector (which is good because these changes may be due to error).

Adding this penalty term can help us establish a more robust calibration model that doesn't overfit the data and is less sensitive to random error in the spectral measurements. But it is does not take into account the structure of error due to disturbance variables; in other words, it is not tailored to our specific calibration challenge.

The penalty term can be tailored to address the specific calibration challenge. In showing the variance induced by changes to disturbance variables (Figure A.3) it is clear that there is structure to the disturbance-induced error. That is, the disturbance variables do not induce random error. For example, probe alignment changes tends to shift whole and scale whole regions of the spectum.

The penalty term used to create ridge regression does not take into account this type of structured error: it assumes that the error in each input feature is independent of the error in the other features and also that the error in each feature is on the same order of magnitude. Because we can probe the spectral variance introduced by changing disturbance variables, we can tailor the penalty term.

The disturbance-induced x-error covariance matrix. To characterize how the input features collectively change with changing disturbance variables we can use a covariance matrix. For this specific application, the matrix is termed the disturbance-induced $\mathbf{x}$-error covariance matrix.

To find this matrix, a number of spectra were recorded for the *same* solution while the four identified disturbance variables were systematically changed. (This experiment will be referred to as the Spectral Variance Experiment). Because the solution composition was fixed for all of the measurements made during the Spectral Variance Experiment, the collected data set has the following form:

$$\{(\mathbf{x}^{[1]}, y), \cdots, (\mathbf{x}^{[N]}, y)\},$$

where y is fixed throughout.

From this data set we can estimate the variance induced by the disturbance variables. If the measurements were perfect with no error, then every $\mathbf{x}^{[i]}$ would be exactly the same. We can therefore get a sense of the error in input features by comparing how much each measurement deviates from the average:

$$e(\mathbf{x}^{[i]}) = \mathbf{x}^{[i]} - \bar{\mathbf{x}}.$$

Combined, this data can be used to estimate an error-covariance matrix:

$$\text{let } \Delta X \equiv \left[e(\mathbf{x}^{[1]}), \cdots, e(\mathbf{x}^{[N]}) \right],$$

then the error-covariance matrix is defined as:

$$\Sigma \equiv \Delta X^{\mathrm{T}} \Delta X.$$

The disturbance-induced **x**-error covariance matrix obtained in this way from data collected during the Spectral Variance Experiment is distinguished by the notation: Σ_{SVE}.

Using the disturbance-induced x-error covariance matrix to create a tailored penalty term. Replacing the identify matrix in the ridge-regression penalty term (equation A.7) with Σ_{SVE} gives us a new, better-suited penalty term

$$p(\mathbf{w}, b) = \mathbf{w}^{\mathrm{T}} \Sigma_{\mathrm{SVE}} \mathbf{w} + b^2. \tag{A.9}$$

In minimizing this term, we are identifying an affine calibration model that gives the same, or close to the same, prediction for each of the spectra recorded during the Spectral Variance Experiment. In other words, in minimizing the tailored penalty term we can preference calibration models that are *insensitive* to changes in the spectra caused by the disturbance variables.

The fit function can also be tailored to reduce overfitting. The fit function in least-squares regression is the sum of squared error between the calibration-predicted and the known concentrations across the training data. This is only one of many different fit functions that could be used (although it is especially convenient for computational mathematics).

In addition to adding and then tailoring a penalty function, we can also tailor the fit function to adjust the regression algorithm. One type of fit function that makes some sense when minor error is expected to always persists and is not such a problem, is a so-called dead zone fit function. In this type of fit function, no cost is given to small errors in the model-prediction. For example, a dead zone *linear* fit function is the following:

$$\begin{aligned} &\Phi = \sum_{i=1}^{N} \phi^{[i]}, \\ &\text{where } \phi^{[i]} \equiv \left\{ \begin{array}{ll} 0 & \textbf{if } |\hat{y}^{[i]} - y^{[i]}| \leq \epsilon, \\ |\hat{y}^{[i]} - y^{[i]}| - \epsilon & \textbf{else}, \end{array} \right\} \end{aligned} \tag{A.10}$$

and ϵ is an adjustable parameter that dictates the width of the dead zone.

A.1.6 The Robust Parameter Support Vector Regression Algorithm

Adding the penalty function given by Equation (A.9) to the dead zone linear fit function given by (A.10), creates a regression algorithm that we labeled Robust Parameter Support Vector Regression:

$$\begin{array}{ll} \underset{\mathbf{x},\, b}{\text{minimize}} & \sum_{i=1}^{N} \phi^{[i]} + \sigma(\mathbf{w}^{\mathrm{T}} \Sigma_{\text{SVE}} \mathbf{w} + b^2) \\ \text{subject to} & \phi^{[i]} \equiv \left\{ \begin{array}{ll} 0 & \textbf{if } |\hat{y}^{[i]} - y^{[i]}| \leq \epsilon, \\ |\hat{y}^{[i]} - y^{[i]}| - \epsilon & \textbf{else}, \end{array} \right\} \\ & \hat{y}^{[i]} = \mathbf{w}^{\mathrm{T}} \mathbf{x}^{[i]} + b, \text{ for } i = 1, \ldots, N. \end{array} \tag{A.11}$$

By including the tailored penalty term and adding slack in the fit function, RPSVR is able to better separate the signal from the noise: it produces a calibration model that fits the training data while at the same time being insensitive to errors in the spectrum measurements caused by the disturbance variables.

Additional benefits of the structure of the regression algorithm. The combination of the dead zone linear fit function with the tailored penalty term was not selected randomly. The regression algorithm was formulated in this way to produce a regression algorithm that is very similar to the well-established Support Vector

Regression (SVR). (Simply replacing the penalty term in RPSVR with the standard ridge-regression penalty term gives SVR). Aside from making it easy to name the regression algorithm, having nearly the same structure as SVR has a two key advantages related to computation and fitting. The first is that the optimization problem is convex. This lets us solve the optimization problem rapidly [Boyd and Vandenberghe, 2009]. The second is not as easy to see, but it is a powerful advantage: the specific setup enables the use of the *kernel trick* [Smola and Scholkopf, 2004]. In effect, this enables us to find a calibration model that picks up on the intricacies in the features and thereby fit the data better while at the same time suppressing the risk of overfitting [Smola and Scholkopf, 2004, Abu-Mostafa et al., 2012].

A.1.7 Performance of RPSVR

Robust Parameter Support Vector Regression was compared against a number of established regression algorithms, many of which are common for IR-to-concentration calibration, including:

- Principle Component Regression (PCR),
- Partial Least Squares Regression (PLSR),
- Dead Zone Linear Regression (DZLR),
- (standard) Support Vector Regression (SVR), and
- Ridge Regression (RR).

To compare, a test data set was collected. Calibration models were then constructed using set training data and input features, but with the different listed regression algorithms. Each calibration model was then tested on the test set (not used in training). A summary of these results is displayed in Figure A.4.

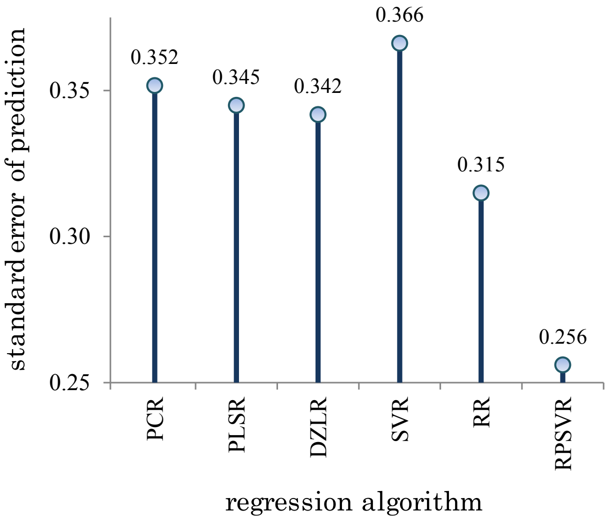

Figure A.4: *Comparison of calibration model accuracy.* Each calibration model was constructed from the same training data set and tested on the same test data set: the differences in prediction accuracy are due to the regression algorithms used.

Judging by accuracy of prediction, the calibration model obtained with RPSVR outperformed each of the other calibration models. Also, comparing RPSVR side-by-side with SVR, we see that the tailored penalty term does in fact make the model less sensitive to the disturbance variables; the RPSVR is more robust, making consistently accurate predictions even as the disturbance variables fluctuate. This is shown in Figure A.5.

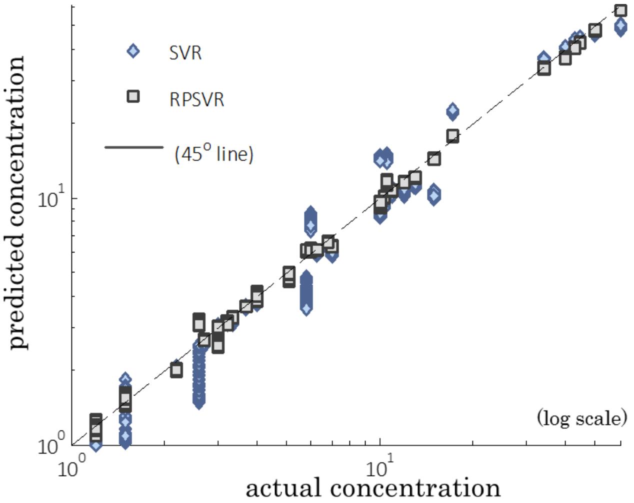

Figure A.5: *Parity plot of calibration model predictions.* Calibration models constructed with standard Support Vector Regression (SVR) and Robust Parameter Support Vector Regression (RPSVR).

A.1.8 Future work with RPSVR

The idea behind Robust Parameter Support Vector Regression can likely find application outside of multicomponent calibration. Highlighted here are two promising research directions.

Calibration maintenance. It is often the case that a calibration model becomes less useful overtime. This is because the conditions under which the model was originally developed are further and further removed from the current conditions and, over time, the instrument may change or be replaced with a new instrument. When this happens, calibration maintenance is required.

The goal of calibration maintenance is to use a small amount of new data to adjust the existing model rather than gather a whole new training set and developing a whole new model. The key idea behind RPSVR—using robust parameter penalty term—is potentially very useful for calibration maintenance. In particular, a small amount of new data can be gathered for a subset of the compositions used in the training data. From this data an error-covariance matrix can be identified that characterizes the differences in the spectral measurements made when the calibration model was originally developed and the spectral measurements made at the current time. This error-covariance matrix can then be used—by way of the penalty term—to identify a model that is insensitive to those measurement changes while still fitting the original training data. Such a model should perform better on new measurements, achieving the goal of calibration maintenance.

Classification: Robust Parameter Support Vector *Machines*. Support Vector *Regression* (SVR) is adapted from the Support Vector Machine (SVM) *classification* algorithm. In this classification algorithm, groups are separated according to a classification line the gives "maximum room for error"—under the assumption that the error is the distributed *equally* across all of the features. An example is shown in Figure A.6(a).

The Robust Parameter penalty term gives us a way of modifying the Support Vector Machine algorithm to find a line that gives maximum room for error when the error is observed to be different for the different features. For example, if the feature x_1 is measured very *in*accurately while the feature x_2 is very accurate, then a different dividing line is better: that shown in Figure A.6(b). Tailoring the penalty function lets us find a dividing line that is more appropriate for the *observed errors* in the features. This could improve accuracy over standard support vector machines.

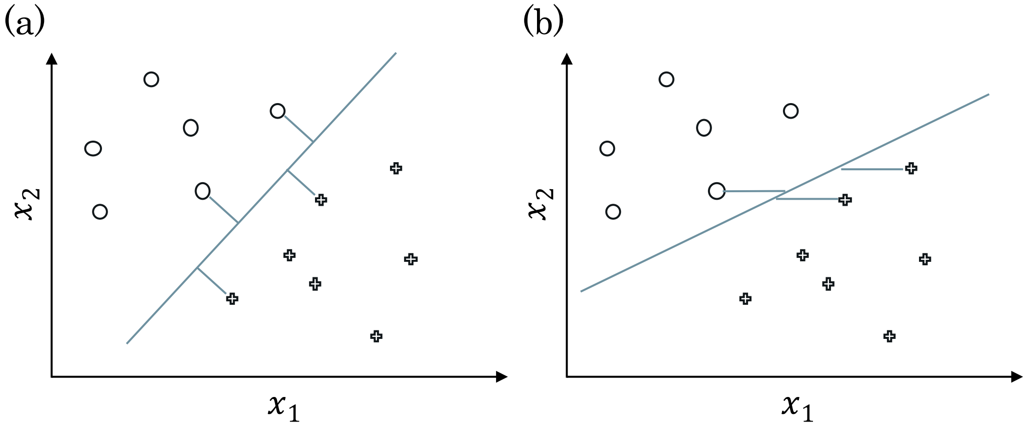

Figure A.6: *Different classification lines.* Two objects mapped according to features x_1 and x_2: circles represent examples of the first object, plus symbols represent examples of the second object. In linear classification, we need to find a dividing line. Figures (a) and (b) illustrate two different dividing lines: (a) the SVM dividing line, (b) the RPSVM dividing line when x_1 is inaccurate.

Additional work on the robust parameter support vector machine algorithm is required. The Support Vector Machine algorithm is one of the most heavily used and widely successful algorithms for classification. A large part of this success can be attributed to the fact that the algorithm facilitates the use of the kernel trick [Abu-Mostafa et al., 2012], which lets us find non-linear dividing lines without the same risk of overfitting. I expect that the Robust Parameter version also admits the application this same type of strategy—and if it does, the adjusted penalty function can be of benefit to classification accuracy. But, I have been unable to mathematically show this is true.

If I had more time in my PhD, I would further explore Robust Parameter Support Vector *Machines* to determine if this idea has utility, whether or not it has already been published, and examine the use of kernels with the adjusted penalty term.

A.2 Molar Supersaturation for Multicomponent Salts

In [Griffin et al., 2015d] a measure termed the molar supersaturation was developed. This measure is analogous to the supersaturation, σ given by Equation (1.4), but for a multicomponent salt that dissociates in solution. The development of molar supersaturation is repeated here.

For the generic crystallization system consisting of solute A (non-dissociating) in water illustrated in Figure A.7, the thermodynamic driving force for crystallization is given by the difference in chemical potential:

$$\Delta\mu^{\text{cryst}} \equiv \mu_{\text{A}}^{\mathbf{(L)}}(x_{\text{A}}, T, P) - \mu_{\text{A}}^{\mathbf{(S)}}(T, P),$$

where $\mu_{\text{A}}^{\mathbf{(L)}}$ is the chemical potential of the solute, A, in solution and x_{A} is the mole fraction of the solute in solution; $\mu_{\text{A}}^{\mathbf{(S)}}$ is the chemical potential of the solute in a pure solid state. (As described in §1.1, but with the mole fraction of the solute given in place of the concentration).

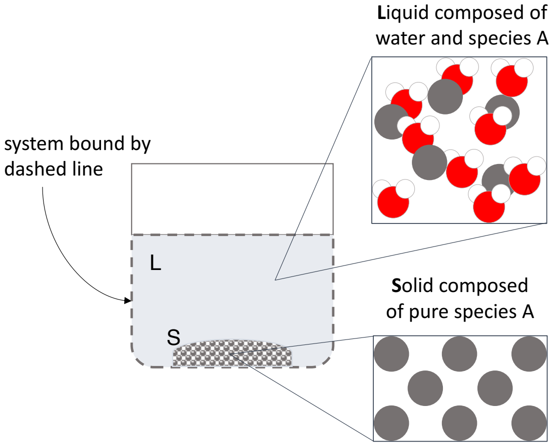

Figure A.7: Illustration of a simple two-phase system: liquid composed of water and solute A (non-dissociating); solid composed of pure solute A.

If a two-phase closed system is held at constant temperature and pressure, eventually an equilibrium state is reached. In this state the composition of each phase is fixed and the chemical potential for a given species is uniform across the phases (ignoring any surface effects). When pure component A in a solid phase is in equilibrium with an aqueous solution containing A, the following expression must hold:

$$\Delta\mu^{\text{cryst}} = 0 = \mu_{\text{A}}^{\textbf{(L)}}(x_{\text{A}}^{*}(T,P),T,P) - \mu_{\text{A}}^{\textbf{(S)}}(T,P),$$

where $x_{\text{A}}^{*}(T,P)$ is the equilibrium solution mole-fraction or *solubility* of solute A at the given temperature and pressure.

Using this relationship, we can re-express the driving force for crystallization:

$$\Delta\mu^{\text{cryst}} = \mu_{\text{A}}^{\textbf{(L)}}(x_{\text{A}},T,P) - \mu_{\text{A}}^{\textbf{(L)}}(x_{\text{A}}^{*}(T,P),T,P). \tag{A.12}$$

At a fixed pressure (typical of cooling crystallizations), the driving force can be measured if the dependency of chemical potential on the solution temperature and composition is known. A classic expression for this dependency is:

$$\mu_{\text{A}}^{\textbf{(L)}}(x_{\text{A}},T,P) = \mu_{\text{A}}^{\circ} + RT\,ln\,(\gamma_{\text{A}}x_{\text{A}})\,,$$

where R is the ideal gas constant, μ_{A}° is the reference state chemical potential for solute A and γ_{A} is the activity coefficient, which is a function of both the solute mole fraction and the temperature.

The driving force for crystallization, as expressed in Equation (A.12), can now be re-written again:

$$\Delta\mu^{\mathrm{cryst}} = \left[\mu^{\circ}_{\mathrm{A}} + RT\,ln\left(\gamma_{\mathrm{A}} x_{\mathrm{A}}\right)\right] - \left[\mu^{\circ}_{\mathrm{A}} + RT\,ln\left(\gamma^{*}_{\mathrm{A}} x^{*}_{\mathrm{A}}(T)\right)\right]$$

rearranging and canceling

$$\Delta\mu^{\mathrm{cryst}} = RT\,ln\left(\frac{\gamma_{\mathrm{A}}}{\gamma^{*}_{\mathrm{A}}}\right) + RT\,ln\left(\frac{x_{\mathrm{A}}}{x^{*}_{\mathrm{A}}(T)}\right).$$

If the activity coefficient is independent of x_{A} or only weakly depends on x_{A}, then for any x_{A} close to the solubility mole fraction, x^{*}_{A}:

$$\frac{\Delta\mu^{\mathrm{cryst}}}{RT} \approx ln\left(\frac{x_{\mathrm{A}}}{x^{*}_{\mathrm{A}}(T)}\right) \approx \frac{x_{\mathrm{A}} - x^{*}_{\mathrm{A}}(T)}{x^{*}_{\mathrm{A}}(T)}, \qquad \text{(A.13)}$$

where the second approximation results from the first-order Taylor expansion.

Equation (A.13) yields the following quantity as measure of the thermodynamic driving force for crystallization of a non-dissociating solute from a single-component solution:

$$\sigma^{[x]}_{\mathrm{A}} \equiv \frac{x_{\mathrm{A}} - x^{*}_{\mathrm{A}}(T)}{x^{*}_{\mathrm{A}}(T)} \left(\approx \frac{\Delta\mu^{\mathrm{cryst}}}{RT} \text{ by above assumptions}\right). \qquad \text{(A.14)}$$

In [Griffin et al., 2015d] this quantity is referred to as the *molar* supersaturation.

In crystallization literature, the amount of solute in solution is often expressed in terms of amount solute per mass solvent. From mole-fractions, concentration (in per mass-solvent units) can be found and vice versa. In these units, the supersaturation

$$\sigma = \frac{C - C^{*}(T)}{C^{*}(T)}$$

is the more common measure (and is the measure that I refer to throughout the thesis).

This expression is closely related to the expression given by Equation (A.14). The reason for going through the trouble of developing $\sigma^{[x]}$ is to make assumptions clear and set the stage for a consistent measure of the driving force for the crystallization of dissociated salts, like darapskite salt.

For multicomponent salts the supersaturation expressed simply by Equation (1.4) may not be an appropriate expression for estimating the driving force for crystallization. Instead of the concentration of a single solute in solution, it is solution composition in terms of dissociated ions that must considered. Equations are given here for the molar supersaturation of a generic salt, $\mathrm{A}_{\nu_\mathrm{A}}\mathrm{B}_{\nu_\mathrm{B}}$, and also for darapskite.

Molar supersaturation for a generic salt, $\mathbf{A}_{\nu_\mathbf{A}}\mathbf{B}_{\nu_\mathbf{B}}$. For a generic salt of the form $\mathrm{A}_{\nu_\mathrm{A}}\mathrm{B}_{\nu_\mathrm{B}}$, the equilibrium between the solid phase and the dissociated ions in solution can be expressed as follows:

$$\underbrace{\mathrm{A}_{\nu_\mathrm{A}}\mathrm{B}_{\nu_\mathrm{B}}}_{\text{solid salt}} \rightleftharpoons \underbrace{\nu_\mathrm{A}\mathrm{A} + \nu_\mathrm{B}\mathrm{B}}_{\text{ions in solution}} \ .$$

A difference in chemical potential of the ions in solution and the solid salt provides the driving force for crystallization (dissolution):

$$\Delta\mu_{\ \mathrm{A}_{\nu_\mathrm{A}}\mathrm{B}_{\nu_\mathrm{B}}} = \underbrace{\nu_\mathrm{A}\mu^\mathrm{L}_\mathrm{A}\left(T, x_\mathrm{A}, x_\mathrm{B}\right) + \nu_\mathrm{B}\mu^\mathrm{L}_\mathrm{B}\left(T, x_\mathrm{A}, x_\mathrm{B}\right)}_{\text{chemical potential of ions in solution}} + \underbrace{\mu^S_{\mathrm{A}_{\nu_\mathrm{A}}\mathrm{B}_{\nu_\mathrm{B}}}(T)}_{\text{chemical potential of solid salt}} \ .$$

Following the same line of arguments used to develop $\sigma^{[x]}$, but this time for the multicomponent salt, yields the following expression:

$$\sigma^{[x]}_{\mathrm{A}_{\nu_\mathrm{A}}\mathrm{B}_{\nu_\mathrm{B}}} \equiv \frac{x_\mathrm{A}^{\nu_\mathrm{A}} x_\mathrm{B}^{\nu_\mathrm{B}} - \left(x^*_\mathrm{A}(T)\right)^{\nu_\mathrm{A}}\left(x^*_\mathrm{B}(T)\right)^{\nu_\mathrm{B}}}{\left(x^*_\mathrm{A}(T)\right)^{\nu_\mathrm{A}}\left(x^*_\mathrm{B}(T)\right)^{\nu_\mathrm{B}}}.$$

From this expression we see that the molar supersaturation is expressed in terms of the dissociated ions in solution rather than a single solute concentration.

Molar supersaturation expression for darapskite. For crystallization of darapskite ($\mathrm{Na_3NO_3SO_4 \cdot H_2O}$), the equilibrium expression is

$$\underbrace{\mathrm{Na_3(NO_3)(SO_4) \cdot H_2O}}_{\text{solid salt}} \rightleftharpoons \underbrace{3\mathrm{Na}^+ + \mathrm{NO}_3^- + \mathrm{SO}_4^{2-} + \mathrm{H_2O}}_{\text{components in solution}},$$

and the molar supersaturation is written as

$$\sigma^{[x]}_{\mathrm{Na_3NO_3SO_4 \cdot H_2O}} \equiv \frac{\left((x_{\mathrm{Na^+}})^3\, x_{\mathrm{NO_3^-}} x_{\mathrm{SO_4^{2-}}} x_{\mathrm{H_2O}}\right) - \left((x^*_{\mathrm{Na^+}})^3\, x^*_{\mathrm{NO_3^-}} x^*_{\mathrm{SO_4^{2-}}} x^*_{\mathrm{H_2O}}\right)}{(x^*_{\mathrm{Na^+}})^3\, x^*_{\mathrm{NO_3^-}} x^*_{\mathrm{SO_4^{2-}}} x^*_{\mathrm{H_2O}}}, \qquad (A.15)$$

where each solubility mole fraction (x^*) is a function of temperature.

From a the closed mass balance and stoichiometry, the composition of the solution for System 1 (§4.1.1), can be specified by the mole fraction of nitrate ($\mathrm{NO_3^-}$) and sulfate ($\mathrm{SO_4^{2-}}$). In other words, $x_{\mathrm{Na^+}}$ and $x_{\mathrm{H_2O}}$ can be written as functions of $x_{\mathrm{NO_3^-}}$ and $x_{\mathrm{SO_4^{2-}}}$. Moreover, the mole fractions of nitrate and sulfate can be inferred from solution concentration monitoring made via ATR-FTIR (§4.3 & §A.1). The end result is that the molar supersaturation of the darapskite in the model solution is just a function of the concentration of nitrate and sulfate and the solubility curve:

$$\underbrace{\sigma^{[x]}_{\mathrm{Na_3NO_3SO_4 \cdot H_2O}}}_{\text{supersaturation darapskite}} = \sigma^{[x]}_{\mathrm{Na_3NO_3SO_4 \cdot H_2O}} \left(C_{\mathrm{NO_3^-}}, C_{\mathrm{SO_4^{2-}}}, \mathbf{C}^*_{\mathrm{Na_3NO_3SO_4 \cdot H_2O}}(T)\right), \qquad (A.16)$$

where $\mathbf{C}^*_{\mathrm{Na_3NO_3SO_4 \cdot H_2O}}(T)$ is the solubility curve of darapskite obtained from the solubility trace methodology.

A.3 Solubility Trace Methodology for Multicomponenet Solutions

For a simple solution (non-dissociating solute in a solvent), the solubility is the amount of solute dissolved when the two-phase system—pure solute in the solid phase, solute and solvent in the liquid phase—is in equilibrium at a fixed temperature and pressure. Thus, for a single-solute system at fixed pressure, the solubility can be expressed in terms of the solute concentration as a function of temperature (as given in §1.1 and shown in Figure 1.2):

$$C^* = C^*(T) \text{single-solute solution, fixed pressure.}$$

When multiple electrolytes exist in a solution, finding the solubility of a component becomes more complex; multiple solid phases may be formed from the ions in the solution, and the solubility of each salt is dependent on the concentration of all components in the solution. Fully characterizing the solubility of a salt in a multicomponent electrolytic system is therefore an experimentally arduous task, requiring equilibrium solution composition data at varying temperatures and overall system compositions. Fortunately, for batch-cooling crystallizations, the system is closed and maintained at a constant pressure. Thus only the temperature dependence of solubility of the crystallizing salt must be characterized to apply supersaturation control. Even so, this must be done for each new batch if the feed composition changes (i.e. each different overall system composition). Therefore a methodology for efficiently collecting the requisite solubility data is desired.

An experimental procedure referred to as the polythermal method [Mullin, 2001] is commonly employed to determine the solubility as a function of temperature prior to controlled crystallization operations. This method does not require sophisticated measurement techniques but does necessitate multiple experiments. Alternatively, we propose taking advantage of *in situ* solution composition monitoring to more efficiently collect the solubility data. This new method is referred to as the solubility trace method.

The polythermal method. For single-solute solutions, the solubility as a function of temperature is often identified with a series of cooling/heating operations at varying solute concentration levels [Mullin, 2001, Fujiwara et al., 2002, Barrett and Glennon, 2002]. This method has been termed the polythermal method. Starting with an undersaturated solution of known composition, the temperature is reduced until nucleation is observed. Once crystals have formed, the temperature is increased until complete dissolution is observed. This marks a single point on the solubility curve. The procedure is illustrated in Figure A.8.

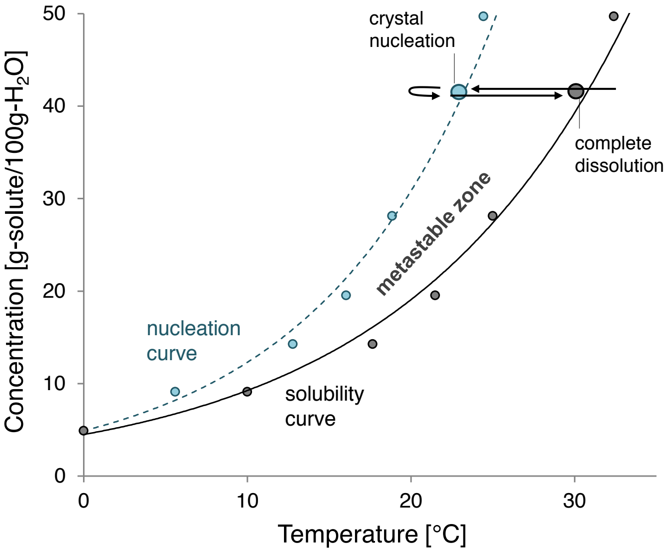

Figure A.8: *Solubility curve identified by the polythermal method.* The arrows suggest the order of operations: at each selected initial concentration, the solution is slowly cooled until nucleation is observed (marking a point on the nucleation curve); the solution and suspended crystals are then reheated until complete dissolution is observed (marking a point on the solubility curve).

There are a number of variables that can affect the degree of supersaturation that builds before nucleation is observed (i.e. the width of the metastable zone) including rate of temperature change, impurities, and even the solution history. The temperature at which complete dissolution is observed is less variable, but can be affected by the heating rate. To obtain an accurate estimate of the solubility curve, the heating rate must be slow enough such that solid-liquid equilibrium is approximately maintained.

The solubility trace method. The solubility trace method is an alternative to the polythermal method that takes advantage of on-line composition monitoring and collects solubility data in a single experiment. A number of strategies similar to the solubility trace methodology presented here have been previously discussed or implemented [Woo et al., 2009, Barrett et al., 2010, Saleemi et al., 2012].

For simplicity, the solubility trace methodology is introduced for the simple solution containing sodium nitrate ($NaNO_3$) and water. The solubility of sodium nitrate in water can be found by the polythermal method as shown in Figure A.9(a).

This, however, requires multiple experiments. Alternatively, the solubility curve can be found more directly by applying *in situ* concentration monitoring during a single cooling/heating operation. The technique is illustrated for sodium nitrate in Figure A.9(b). Cooling a solution initially containing 120 g-$NaNO_3$/100 g-water causes supersaturation to build until around 52.5 °C, at which point nucleation and rapid crystallization is observed; on heating the solid-liquid slurry, the measured solution concentration of sodium nitrate traces out the solubility curve over the temperature range probed.

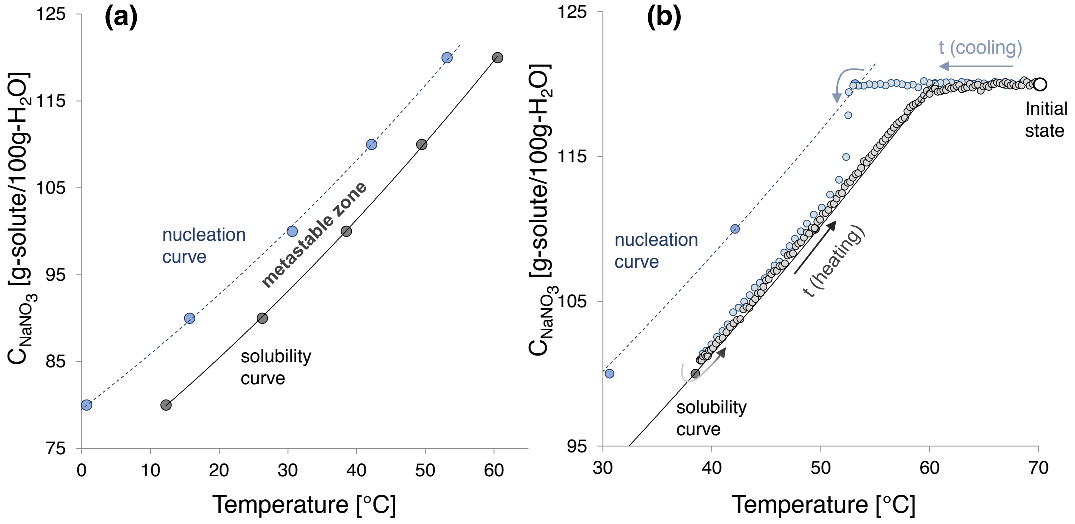

Figure A.9: *Slubility curve for sodium nitrate identified by the solubility trace methodology.* (a) Shows the curves obtained using the polythermal method; while (b) shows the solubility curve for temperatures above 40 °C can obtained by the solubility trace method (solubility curve is traced out by the concentration-temperature profile during heating stage)

The same solubility trace strategy can be employed to identify the solubility in a multicomponent solution. That is, for a clear solution of a given composition, a cooling-heating operation can be coupled with composition monitoring to trace out the solubility for a given salt within a specified temperature range. It is important to note that such a curve is unique to the initial composition probed; different solubility curves are expected for different initial compositions.

Depending on the number of independent components in solution, the solubility curve takes on additional dimensions. For the darapskite crystallizing from the multicomponent electrolytic solution, the curve takes on a third dimension—with coordinates of temperature, concentration of nitrate, and concentration of sulfate:

$$\mathbf{C}^*_{\mathrm{Na_3NO_3SO_4\cdot H_2O}}(T) = \Big[C^*_{\mathrm{NO_3^-}}(T)\ C^*_{\mathrm{SO_4^{2-}}}(T)\Big]\,.$$

Figure A.10 shows the solubility trace for darapskite crystallizing from a solution initially composed of 330 grams of $NaNO_3$ and 21.75 grams of Na_2SO in 300 mL of water: cooling drives the crystallization of darapskite; these crystals are then dissolved during heating and the profile during dissolution (grey) traces out the solubility curve. Once the data is collected, the functional dependence of temperature can be captured with a polynomial fit to the dissolution curve.

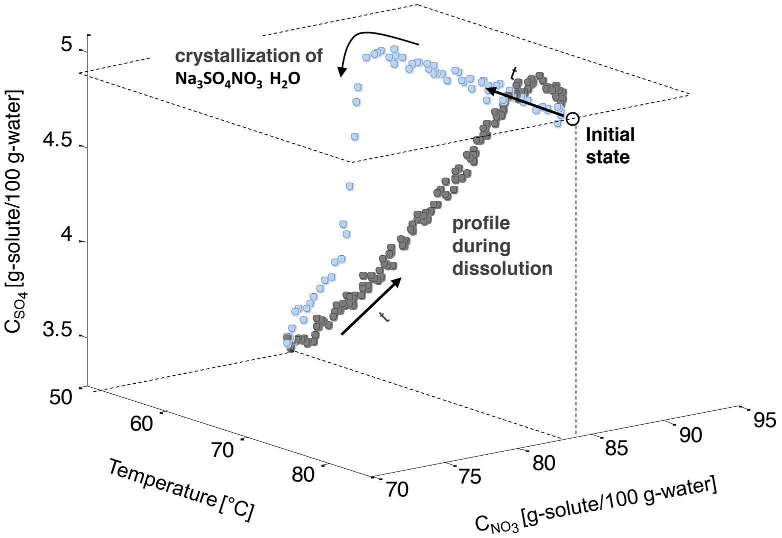

Figure A.10: *Solubility trace for darapskite.* Profile in grey identifies the solubility curve.

APPENDIX B

SIEVE MEASUREMENTS AND THE MASS-PER-COUNT–SIZE RELATIONSHIP

In introducing the MC framework, a measure of the mean crystal size was proposed (Equation 3.4):

$$\bar{L} \equiv \left(\frac{1}{k_\mathrm{S}\rho_\mathrm{crys}}\right)^{1/3}\left(\frac{m}{n}\right)^{1/3},$$

where k_S is a constant shape factor, ρ_crys is the density of the crystal phase, m is the crystal mass, and n is the crystal count (that is, the total number of crystals).

This measure of average size is convenient: assuming the chord count is proportional to the number of crystals, we can obtain a measure of it online

$$\bar{L} = \left(\frac{1}{k_\mathrm{S}\rho_\mathrm{crys}}\right)^{1/3}\left(\frac{m}{n}\right)^{1/3} \propto \left(\frac{m}{c}\right)^{1/3},$$

where c is the chord count measured by FBRM.

Ideally, we would use sieve analysis to directly obtain an independent measure of $\bar{L}$. However, sieve analysis gives the mass in each bin, m_i, not the number of crystals in each bin, n_i. To obtain a measure of $\bar{L}$ from sieve analysis we need to approximate the number of crystals from the mass data. This can be done with an assumption about the mean volume of the crystals in each sieve bin.

Let $\bar{v}_i$ be a measure of the mean volume of a crystal in the i^th sieve bin, defined as:

$$\bar{v}_i \equiv \left(\frac{m_i/\rho_\mathrm{crys}}{n_i}\right). \tag{B.1}$$

This mean volume should be related to the bin midpoint, s_i—i.e. the midpoint of the size range for the i^th sieve bin.

Here, we will *assume* that the average volume of a crystal in the i^{th} bin is proportional to the cube of the bin midpoint and that the following relation holds:

$$\bar{v}_i = k_{\text{S}} s_i^3, \tag{B.2}$$

where we k_{S} is the same shape factor we introduce in defining $\bar{L}$.

Rearranging Equation (B.1) we get

$$n_i = \left(\frac{m_i/\rho_{\text{crys}}}{\bar{v}_i} \right),$$

and then, using Equation (B.2), we obtain an estimate of the number of crystals per bin from the mass of crystals per bin

$$n_i = \left(\frac{m_i/\rho_{\text{crys}}}{k_{\text{S}} s_i^3} \right). \tag{B.3}$$

Using this approximation, we obtain a measure of $\bar{L}$ from sieve data:

$$\bar{L} = \left(\frac{1}{k_{\text{S}} \rho_{\text{crys}}} \right)^{1/3} \left(\frac{\sum_{i=1}^{N_{\text{bins}}} m_i}{\sum_{i=1}^{N_{\text{bins}}} n_i} \right)^{1/3} = \left(\frac{1}{k_{\text{S}} \rho_{\text{crys}}} \right)^{1/3} \left(\frac{\sum_{i=1}^{N_{\text{bins}}} m_i}{\sum_{i=1}^{N_{\text{bins}}} \left(\frac{m_i/\rho_{\text{crys}}}{k_{\text{S}} s_i^3} \right)} \right)^{1/3}.$$

Canceling out the constants to express this more succinctly:

$$\bar{L} = \left(\frac{m}{\sum_{i=1}^{N_{\text{bins}}} \left(m_i s_i^{-3} \right)} \right)^{1/3}. \tag{B.4}$$

So we see that $\bar{L}$ can be approximated with sieve analysis, which can be experimentally verified: Figure B.1 compares $\bar{L}$ estimated by sieve for paracetamol crystals against the mass-per-count measured online by ATR-FTIR and FBRM.

But measuring the number-based mean size, $\bar{L}$, can be problematic in practice, especially for darapskite crystals. The reason is the sensitivity of this measure to mass in the lowest sieve trays. Figure B.2 illustrates the wide range of aperture sizes in the sieve stack. Comparing the top and bottom sieve trays: $s_{\text{bottom}}^3 << s_{\text{top}}^3$. This means, even a very small mass of crystals in the bottom tray is interpreted as a very large number of crystals. Having such sensitivity would be OK if the measurements were

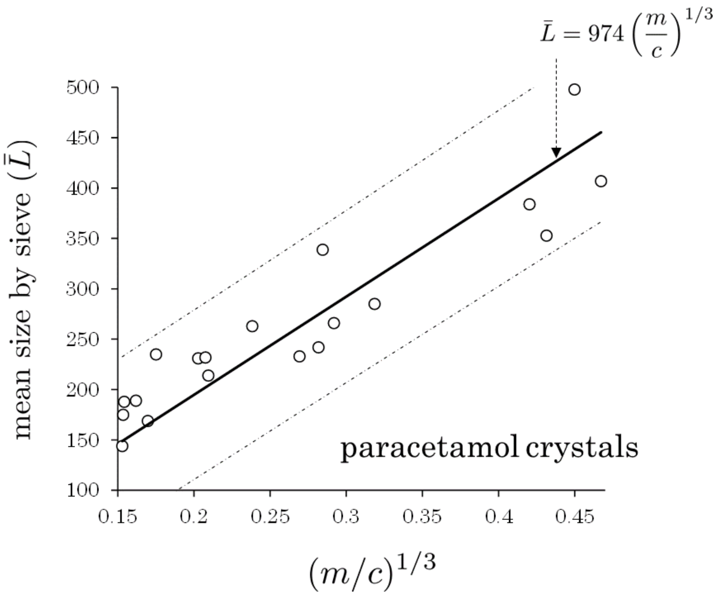

Figure B.1: Mass-per-count–$\bar{L}$ relationship observed for paracetamol crystals.

perfect. Unfortunately, the sieve measurements are not perfect: darapskite crystals are fragile and break in the RO-TAP, resulting in non-negligible amounts of crystal fragments in the lower sieve bins and significantly increasing the estimate of n_i and decreasing the estimate of $\bar{L}$.

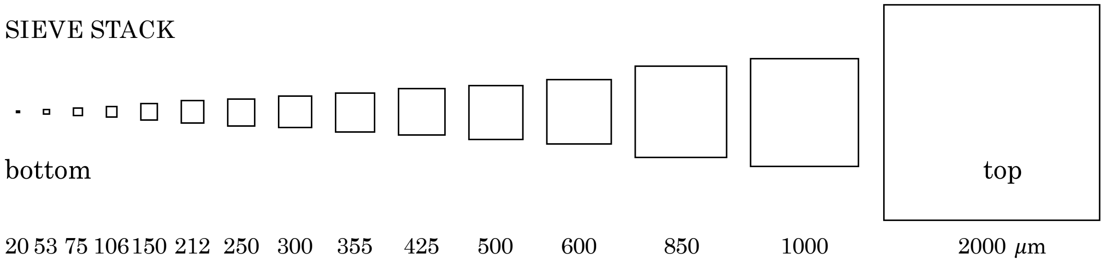

Figure B.2: Illustration of the size of the apertures of each sieve tray in the stack.

To avoid inaccuracies in the sieve measurement, the *mass*-weighted mean crystal size has been used throughout this thesis. In this case, the relationship between the mass and count measured online and mass-weighted mean size measured offline is empirical (c.f. Figures 4.7 and 4.8).

APPENDIX C

COMPUTATION: MARKOV STATE MODEL

C.1 Machine Learning to Obtain an MC Dynamics Model

C.1.1 Training data

The training data for learning a mathematical model of MC space movement for darapskite consists of 6162 $(\mathbf{x}, u)$–$\Delta\mathbf{x}$ data vector pairs, collected over 20 crystallization runs. The training data for learning a mathematical model of MC space movement for paracetamol consists of 7402 $(\mathbf{x}, u)$–$\Delta\mathbf{x}$ data vector pairs collected over 14 crystallization runs.

For both systems, there is a characteristic of the training data that is worth pointing out: the data does not fill the input space fully or evenly. This can be seen in Figure 5.13, and is even more evident when the training data is also spread out according to the supersaturation level u. The key to learning from data with this structure, was using the appropriate weighting scheme in the regression algorithm. This is discussed subsequently.

C.1.2 'Non-parametric' modeling strategy and the weight expression

The mathematical model we obtain from the proposed learning method is non-parametric with respect to position $\mathbf{x}$. That is, instead of identifying a fixed global model that specifies the dynamics for any given $\mathbf{x}$ and u, we keep the training data on hand and obtain a local model for each queried state, $\acute{\mathbf{x}}$. Each local model then specifies the dynamics for any u but only in the neighborhood of the queried state, $\acute{\mathbf{x}}$. The queried state influences the local model through the weight function:

$$w(\acute{\mathbf{x}}, \hat{\mathbf{x}}_j; \kappa) = exp\left(-\frac{||\acute{\mathbf{x}} - \hat{\mathbf{x}}_j||_2^2}{\kappa}\right),$$

where κ is an adjustable parameter termed the bandwidth.

By adjusting the bandwidth we can adjust the sensitivity of the local model to the crystal state. And this gives us a way to learn a non-parametric model from data that is sparse and unevenly spread out in the input space.

Case-based illustration of the local data-weighting strategy. As an illustration, it is informative to look at the weighting schemes used to learn local models for different queried crystal states using different bandwidths (using training data for darapskite). Three cases are enough to illustrate the main point:

Case a: $\acute{\mathbf{x}} = (100, 4)$ using the bandwidth $\kappa = 5$;

Case b: $\acute{\mathbf{x}} = (200, 8)$ using the same bandwidth $\kappa = 5$; and

Case c: $\acute{\mathbf{x}} = (200, 8)$ using a broader bandwidth $\kappa = 10$.

For each different case, the training data points are given different weight—this provides different *effective* training data from which the local models are learned. This is illustrated in Figure C.1 for the three highlighted cases.

From this figure, we see the effect of the weighting scheme: the weighting scheme allows local models to be learned from data collected in the same neighborhood, with the size of the neighborhood determined by the selected bandwidth, κ.

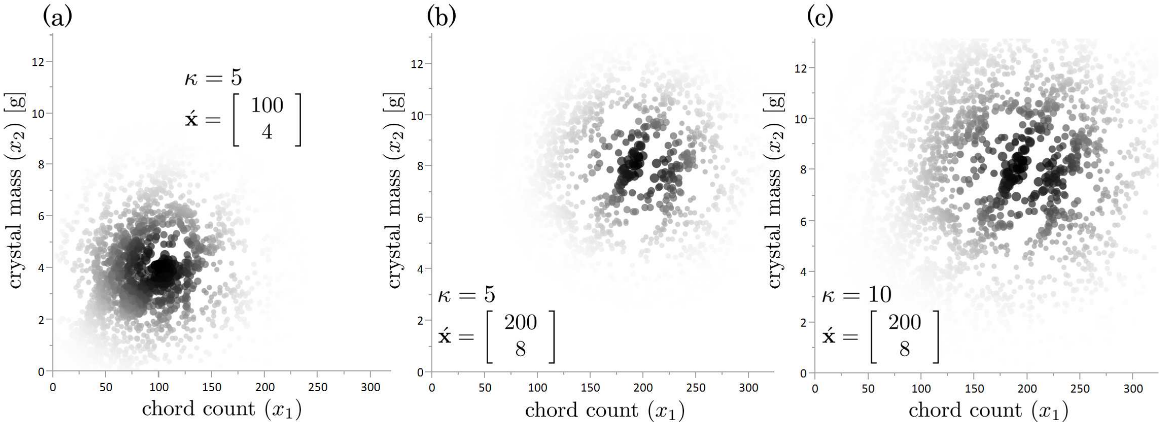

Figure C.1: Visual representation of the weighted training data associated with the example cases a, b, and c. The weight assigned to each training data point is represented by the size and shade of the points.

Scaling the bandwidth lets us learn from sparse, uneven data. In choosing the bandwidth to use at different positions, we have to be cognizant of the data density, which in non-uniform. If we choose a small bandwidth and the queried position is in a data-sparse region, the model may be overly influenced by measurement noise in the few data points that happen to be in that region (i.e. we are in danger of overfitting).

To avoid this the bandwidth is scaled according to the data-density around each queried position in mass-count space. In particular, for each queried point, $\acute{\mathbf{x}}$ we adjusted the bandwidth according to the following expression:

$$\kappa|_{\acute{\mathbf{x}}} = \underset{\kappa}{\operatorname{argmin}} \left(\left(\sum_{j=1}^{N_{\text{train}}} w(\acute{\mathbf{x}}, \hat{\mathbf{x}}_j, \kappa)/N_{\text{train}} \right) - f_{\text{W}} \right)^2 ,$$

where f_{W} is an adjustable parameter that we set to 0.25.

In selecting the bandwidth by this method we ensure that, at each queried point the sum of weights—which range from 0 to 1—amounts to 25% of the training data.

C.1.3 Solving the learning algorithm

$$\underset{\beta \in \mathbf{R}^{2\times 6}}{\text{minimize}} \left(\sum_{j=1}^{N} w(\acute{\mathbf{x}}, \hat{\mathbf{x}}_j; \kappa) || [\hat{u}_j, \hat{u}_j^2, \ldots, \hat{u}_j^6] \beta \Delta t - \Delta \hat{\mathbf{x}}_j ||_2^2 \right)$$

$$\text{subject to} \quad u[u\ u^2 \ldots\ u^6]\beta \succeq \mathbf{0}, \text{ for all } u \in \mathcal{U}_{\mathcal{D}};$$

$$[2u\ 3u^2\ 4u^3\ 5u^4\ 6u^5] \begin{bmatrix} \beta_{2,1} & \beta_{2,2} \\ \beta_{3,1} & \beta_{3,2} \\ \beta_{4,1} & \beta_{4,2} \\ \beta_{5,1} & \beta_{5,2} \\ \beta_{6,1} & \beta_{6,2} \end{bmatrix} \succeq \mathbf{0}, \text{ for all } u \in \mathcal{U}_{\mathcal{D}}.$$

The constraints in the above optimization problem represent the practical implementations of the constraints $uF(\mathbf{x}, u) \succeq \mathbf{0}$ and $\partial F/\partial u \succeq \mathbf{0}$. In these conditions, $\mathcal{U}_{\mathcal{D}}$

denotes the supersaturation domain of interest discretized with a constant difference of 0.001:

$$\mathcal{U}_{\mathcal{D}} = \{-0.150, -0.149, \ldots, 0.500\}$$

In writing the constraints in this way, we obtain an optimization problem that is convex and can be solved in short order using CVX: MATLAB software for convex optimization [Boyd and Vandenberghe, 2009, Grant and Boyd, 2012].

C.2 MC Model Calculations: Optimal Open-Loop Policies, Reachability Analyses, and Stochastic Simulations

The MC model for crystallization and dissolution, expressed mathematically in equation (5.1), can be used to identify optimal open-loop control policies, answer design questions through reachability analyses, and run stochastic simulations.

Optimal open-loop policies are shown for both darapskite and paracetamol. However, as the MC model for darapskite crystallization and dissolution captures reality better than the MC model for paracetamol crystallization and dissolution, the model-based strategy was utilized more fully for darapskite—reachability analyses and stochastic simulations are therefore demonstrated just for darapskite.

C.2.1 Optimal open loop policy calculations.

Our primary control objective is to reach a target final mass-count (and, correspondingly, produce crystals of a target mean size). Additionally, we would like to run the operation with minimal energy input. The collective optimal open-loop control objective can be formalized as:

$$\begin{aligned} &\underset{u_0, \ldots, u_{N_\tau - 1}}{\text{minimize}} \quad \left\{ \rho \sum_{\tau=0}^{N-1} \varepsilon(u_\tau) + d(\mathbf{x}_N, \mathbf{x}^{\oplus}) \right\} \\ &\text{subject to} \quad \mathbf{x}_{\tau+1} = F\big(\mathbf{x}_\tau, u_\tau\big)\Delta t + \mathbf{x}_\tau, \ \tau = 0, \ldots, N-1; \\ &\qquad\qquad\quad \mathbf{x}_0 = \mathbf{x}^{\text{init}}; \end{aligned} \tag{C.1}$$

where N denotes the total number of time intervals and the objective function contains tow terms, a *running cost* and a *terminal cost*:

- the *running cost* is given by the summation of an *input-effort* function $\varepsilon(u) \equiv u^2$, and
- the *terminal cost* is given by the *distance-to-target* function evaluated for the final position, $d(\mathbf{x}_N, \mathbf{x}^{\oplus})$.

Here, the input-effort function reflects the following: crystallization or dissolution will spontaneously bring the system to saturation; maintaining undersaturation or supersaturation during the operation requires a continual temperature change and, consequently, the input of energy. While, the distance-to-target function corresponds to the normalized, squared Euclidean distance between the current MC position and the target MC position [Griffin et al., 2016a].

Solving (C.2.2) ([Griffin et al., 2016a]) yields the optimal open loop supersaturation profile. With the supersaturation profile calculated, the MC model can then be used to obtain the expected MC trajectory under optimal open loop control and the solubility-temperature relationship can be used to obtain the corresponding optimal temperature policy. Optimal open loop policies for different targets were calculated for both darapskite and paracetamol.

Figure C.2 shows the predicted optimal open-loop trajectories for darapskite salt crystallization for three targets. Figure C.3 likewise shows the predicted optimal open-loop trajectories for paracetamol salt crystallization for three targets.

C.2.2 Answering design questions using reachability analyses

Prior to designing a crystallization run, there are general questions we would like to answer, including: What targets are *reachable*? And, what is the *minimum batch time for a given target*?

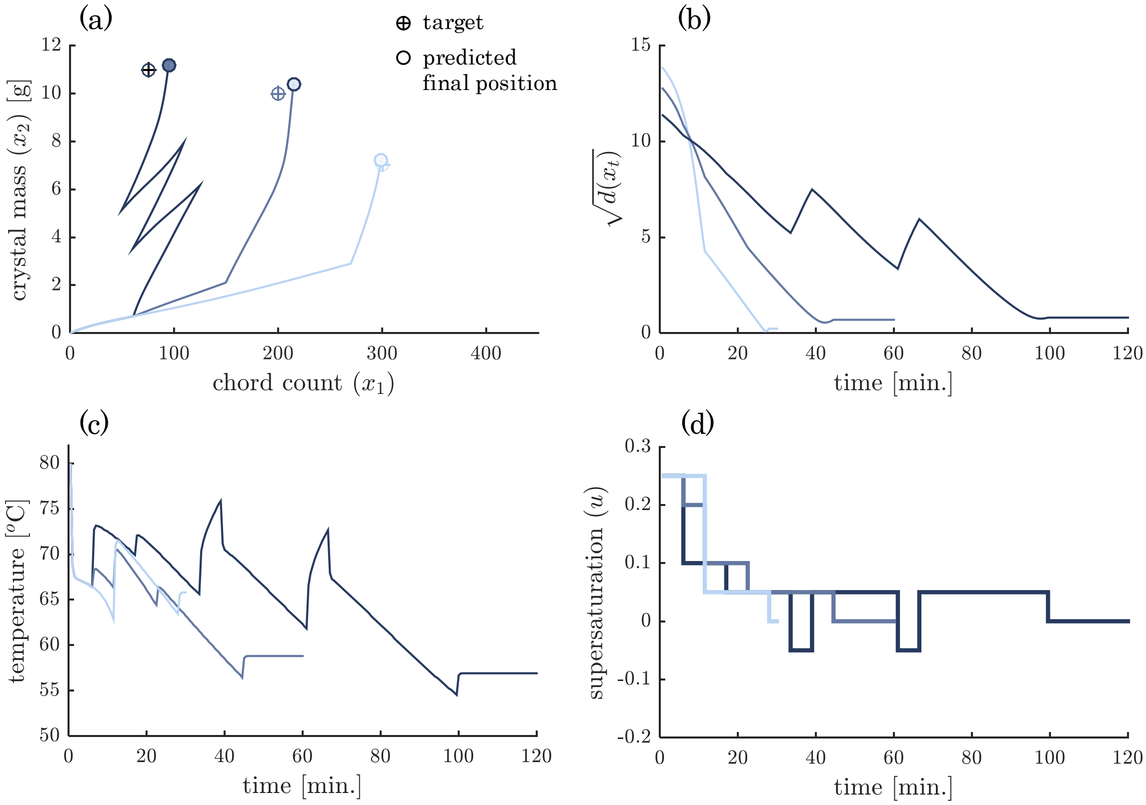

Figure C.2: (a) Predicted trajectories for darapskite crystallization under optimal open loop control towards three separate targets. (b) Predicted distance-to-target profiles for each optimal run. (c) Optimal temperature profiles. (d) Optimal supersaturation profiles.

These questions can be answered conducting reachability analyses. This section illustrates these analyses for darapskite crystallization.

In doing so, two important properties of the system are revealed: a wider range of size targets can be reached in a given batch time by appropriately applying dissolution stages, and there is a nearly linear relationship between the target crystal size and the minimum batch time required.

Reachable regions, darapskite. We aim to understand what regions (in MC space) can be reached during batch cooling crystallization of darapskite salt from water. In particular, we pose two versions of a reachability question:

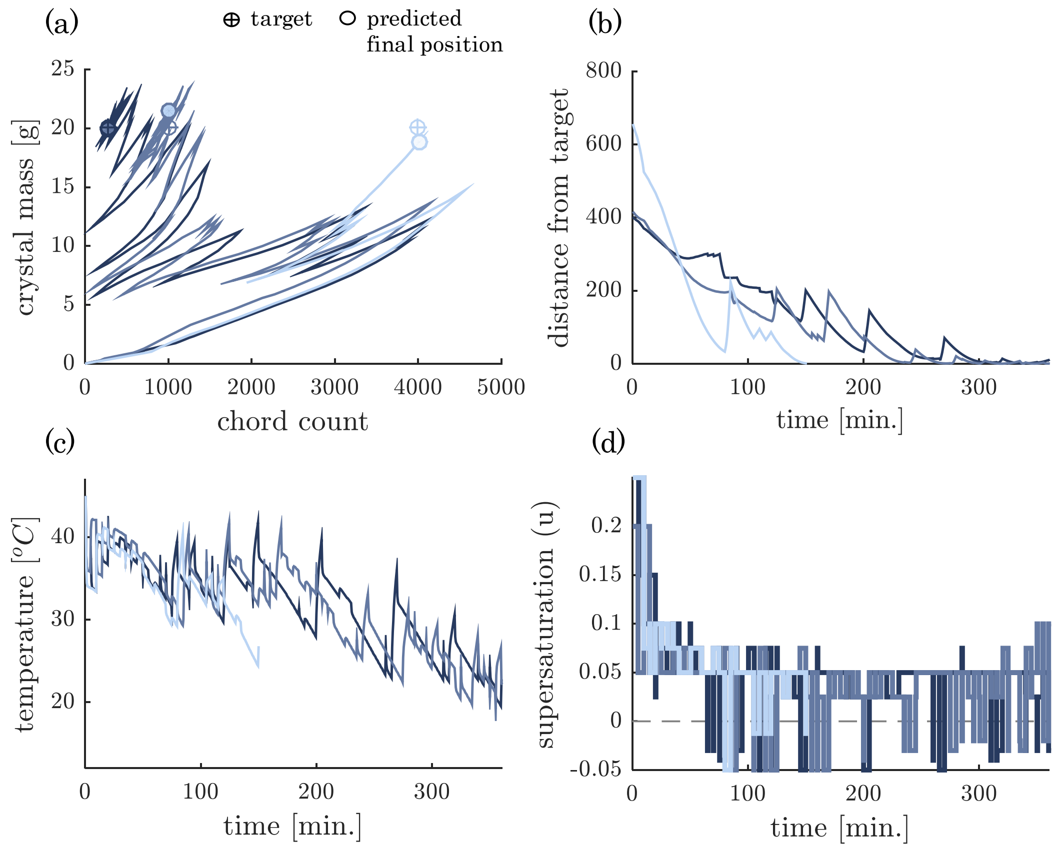

Figure C.3: (a) Predicted trajectories for paracetamol crystallization under optimal open loop control towards three separate targets. (b) Predicted distance-to-target profiles for each optimal run. (c) Optimal temperature profiles. (d) Optimal supersaturation profiles.

a What mass-count (size-yield) targets can be reached applying a positive supersaturation setpoint profile (no dissolution) within a batch time of 150 minutes?

b And, what mass-count targets can be reached in the same batch time if negative supersaturation is allowed?

The model gives us a way to obtain practically usable, albeit approximate answers to these questions. In particular, the model can be used to determine whether or not *specific* targets are reachable by an allowed series of inputs. The reachable region is then approximately mapped out in probing a grid of targets:

$$\mathcal{X}^{\oplus} = \left\{ \mathbf{x}^{\oplus} | x_1^{\oplus} = \{50, 100, \ldots, 350\};\ x_2^{\oplus} = \{2, 4, \ldots, 14\} \right\}$$

To address Question a, we posed the following optimization problem for each target in the grid:

$$\begin{aligned} \underset{u_1,\ldots,u_{N-1}\,\in\,\mathcal{U}_+}{\text{minimize}} \quad & \{d(\mathbf{x}_N, \mathbf{x}^{\oplus})\} \\ \text{subject to} \quad & \mathbf{x}_{\tau+1} = F\big(\mathbf{x}_\tau, u_\tau\big)\Delta t + \mathbf{x}_\tau,\ \ \tau = 0,\ldots,N-1; \\ & \mathbf{x}_0 = [0\ 0]^\top; \end{aligned} \tag{C.2}$$

where $\mathcal{U}_+$ denotes a discrete set of allowed levels of positive supersaturation: $\{0, 0.05, \ldots, 0.25\}$. In solving this optimization problem, the state-space is discretized (c.f. §D.2), the discrete time step is set to 5 minutes, and N is set to 30, such that $t_N = 150$ minutes.

To address Question b., we pose almost the same optimization problem for each target in the grid. The one difference is the allowed the supersaturation setpoints: In this second case, we allow the input to take both positive and negative values

$$\begin{aligned} \underset{u_1,\ldots,u_{N-1}\,\in\,\mathcal{U}}{\text{minimize}} \quad & \{d(\mathbf{x}_N, \mathbf{x}^{\oplus})\} \\ \text{subject to} \quad & \mathbf{x}_{\tau+1} = F\big(\mathbf{x}_\tau, u_\tau\big)\Delta t + \mathbf{x}_\tau,\ \ \tau = 0,\ldots,N-1; \\ & \mathbf{x}_0 = [0\ 0]^\top; \end{aligned} \tag{C.3}$$

where $\mathcal{U} = \{-0.15, -0.10, \ldots, 0.25\}$.

Figure C.4 shows the target positions compared with the optimal final positions when the supersaturation is constrained to be positive (Question a.). Most of the target positions are reached (the final position overlays the target). However, some of the targets in the upper-left of the mass-count space (large crystals) and lower-left of the mass-count space (small crystals) could not be reached in 150 minutes applying a strictly positive supersaturation profile.

In contrast, we find that all of the targets in the grid are reachable when undersaturation is allowed. Figure C.5 shows the target positions compared with the optimal final positions obtained by solving (C.3) for every target in the grid. In this figure, the optimal final positions overlay the target position. This reveals an important property of the system: dissolution cycles can be used to achieve a wider range of crystal sizes.

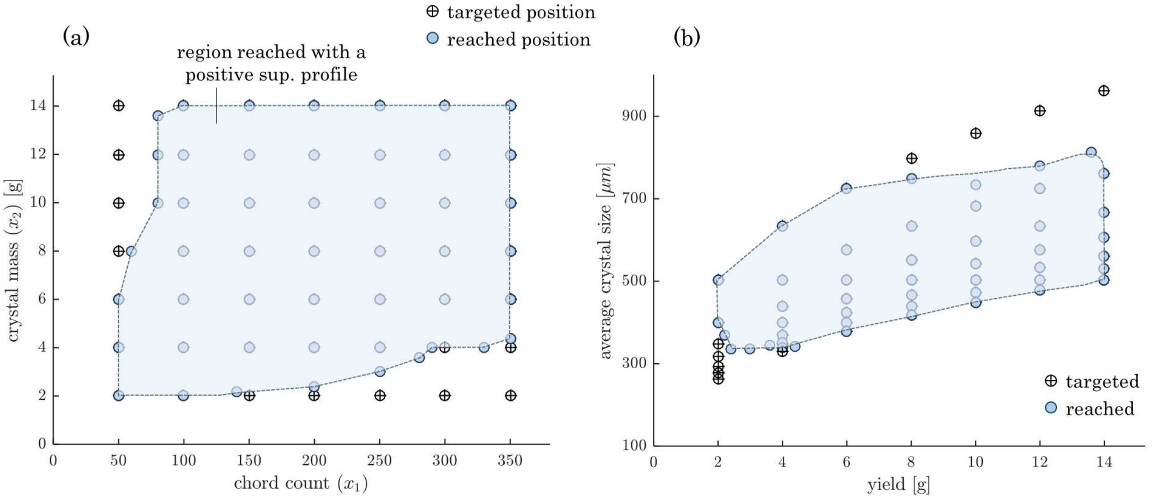

Figure C.4: *Restricting the operation to positive supersaturation restricts the reachable region.* (a) Target mass-count positions compared with optimal final positions when the supersaturation setpoint at each time interval is restricted to be positive or zero and the batch time is set to 150 minutes; positions in the upper-left and lower-right quadrants cannot be reached. (b) Corresponding size-yield targets compared with the reachable size-yield positions.

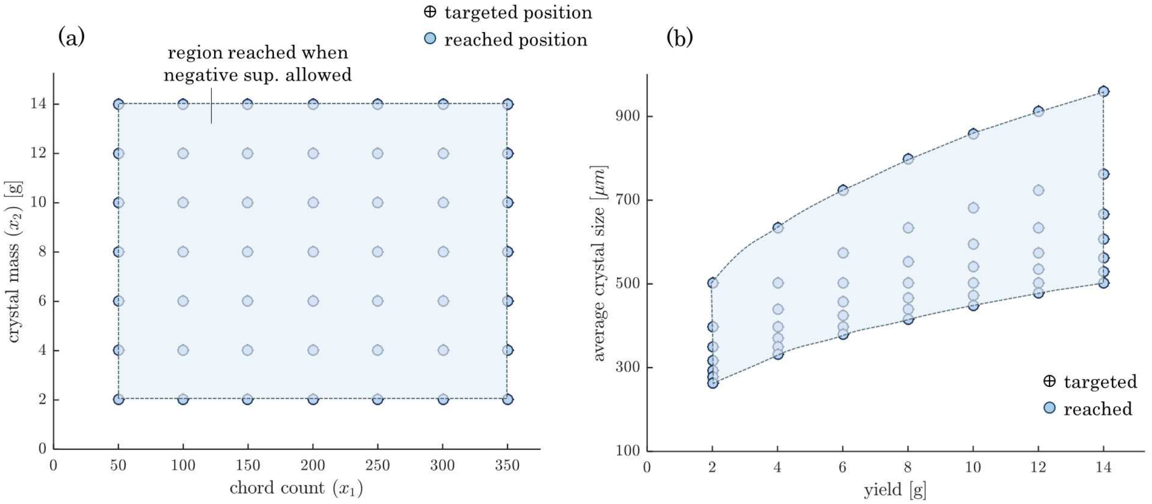

Figure C.5: *Allowing for selective application of undersaturation expands the reachable region.* (a) Target mass-count positions compared with optimal final positions when the supersaturation setpoint during the operation is allowed to take positive or negative values and the batch time is set to 150 minutes; with this more flexible input, each target position can be reached in the allotted batch time. (b) Corresponding size-yield targets compared with the reachable size-yield positions.

Minimum batch times, darapskite. In addition to probing whether or not a target is reachable, it is useful to identify the minimum time required to reach a desired target. For each target the minimum batch time is defined as:

$$\begin{aligned} t_N^*(\mathbf{x}^{\oplus}) \quad = \quad & \text{argmin} \quad \{t_N | t_N = N\Delta t,\ N = 1, 2, ...\} \\ & \text{subject to} \quad d\left(x_N\big(u^*(\mathbf{x}^{\oplus}, N)\big), \mathbf{x}^{\oplus}\right) < \epsilon; \end{aligned} \tag{C.4}$$

where $d\left(x_N\big(u^*(\mathbf{x}^{\oplus}, N)\big), \mathbf{x}^{\oplus}\right)$ is the model-predicted final distance-to-target when the system is under the optimal control profile, $u^*(\mathbf{x}^{\oplus}, N)$, found by solving () for a given target, $\mathbf{x}^{\oplus}$, and batch time defined by N; ϵ represents a specified tolerance for the endpoint-target mismatch, which is set to 1 for this illustration.

Minimum batch times were identified for the targets shown in Figure C.6(a). These times are indicated next to the target and also in Figure C.6(b), which shows the optimal batch time for each of the selected targets against the target average crystal size. This analysis reveals a second important system property: there is a strong, nearly linear tradeoff between the target crystal size and the minimum batch time required.

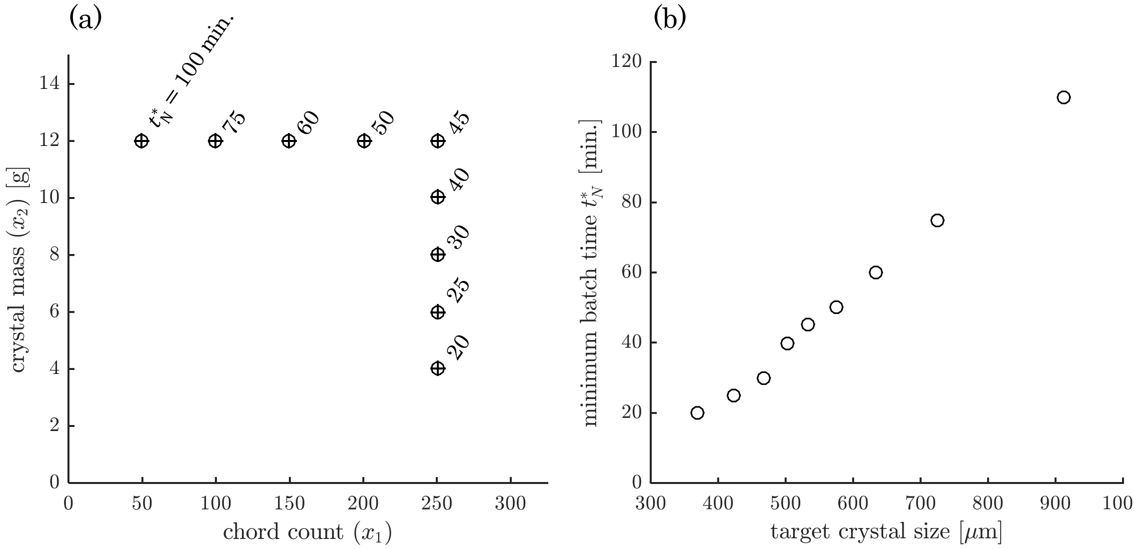

Figure C.6: *Increasing the mean crystal size costs time.* (a) Subset of targets over which the minimum batch time for darapskite crystallization was evaluated; the optimal batch time is shown adjacent to each target. (b) Optimal batch times for selected targets shown against the target mass-weighted mean crystal size of darapskite crystals.

C.2.3 Stochastic simulations, darapskite.

The optimal model-based strategies are developed from a deterministic model. The optimal open-loop control predictions, shown in figures C.2 and C.3, are made under the assumption that the dynamics perfectly align with the model and there is no error or delay in the input. These simulations do not tell us about the expected performance of the state-feedback control when the dynamics are less predictable and there are implementation errors. To probe the feedback control performance we can run stochastic simulations. This is demonstrated for the darapskite system.

To run stochastic simulations, we simulate:

1. Stochasticity in the dynamics

   $$\mathbf{x}_{\tau+1} = F(\mathbf{x}_\tau, u_\tau) + W_\tau,$$

   where W is a random contribution to the dynamics, drawn from an unbiased (zero-mean) multivariate normal distribution:

   $$W_\tau \sim \mathcal{N}\left(\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 430 & 2.32 \\ 2.32 & 0.26 \end{bmatrix}\right).$$

2. Error in the simulated measurements

   $$\hat{\mathbf{x}}_\tau = \mathbf{x}_\tau + e^{\mathbf{x}}_\tau,$$

   and

   $$u^{\mathrm{SP}}_\tau = \pi_\tau(\hat{\mathbf{x}}_\tau),$$

   where $\hat{\mathbf{x}}$ denotes the measured state, $e^{\mathbf{x}}_\tau$ represents random error or noise in the measurement, and u^{SP}_τ is the supersaturation setpoint selected based on the measured state according to the state-feedback policy, π.

The measurement error at each time is drawn from an unbiased multivariate normal distribution with no cross variance

$$e_\tau^{\mathbf{x}} \sim \mathcal{N}\left(\mu = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \Sigma = \begin{bmatrix} 36 & 0 \\ 0 & 0.16 \end{bmatrix}\right).$$

3. Delay and error in the actuation

$$u_\tau = u_{\tau-1} + \delta(u_\tau^{\mathrm{SP}} - u_{\tau-1}) + e_\tau^u,$$

where $\delta \in (0\ 1]$ is a user-specified parameter that adjusts delay in the actuation ($\delta = 1$ corresponds to no delay) and e_τ^u represents random error in the actuation. δ is set to 0.5 and the random actuation error is drawn from an unbiased normal distribution

$$e_\tau^u \sim \mathcal{N}\left(\mu = 0, \sigma = 0.0055\right).$$

The magnitude of the errors and random components used in these stochastic simulations are based in experimental observations ([Griffin et al., 2016a]). However, as it was difficult to be exact in formulating the distributions, we aimed to err on the side of overestimating the random contributions so that we obtained a conservative estimate of the control performance.

Figure C.7 shows a single stochastic simulation of the crystal state trajectories under each of the three feedback policies.

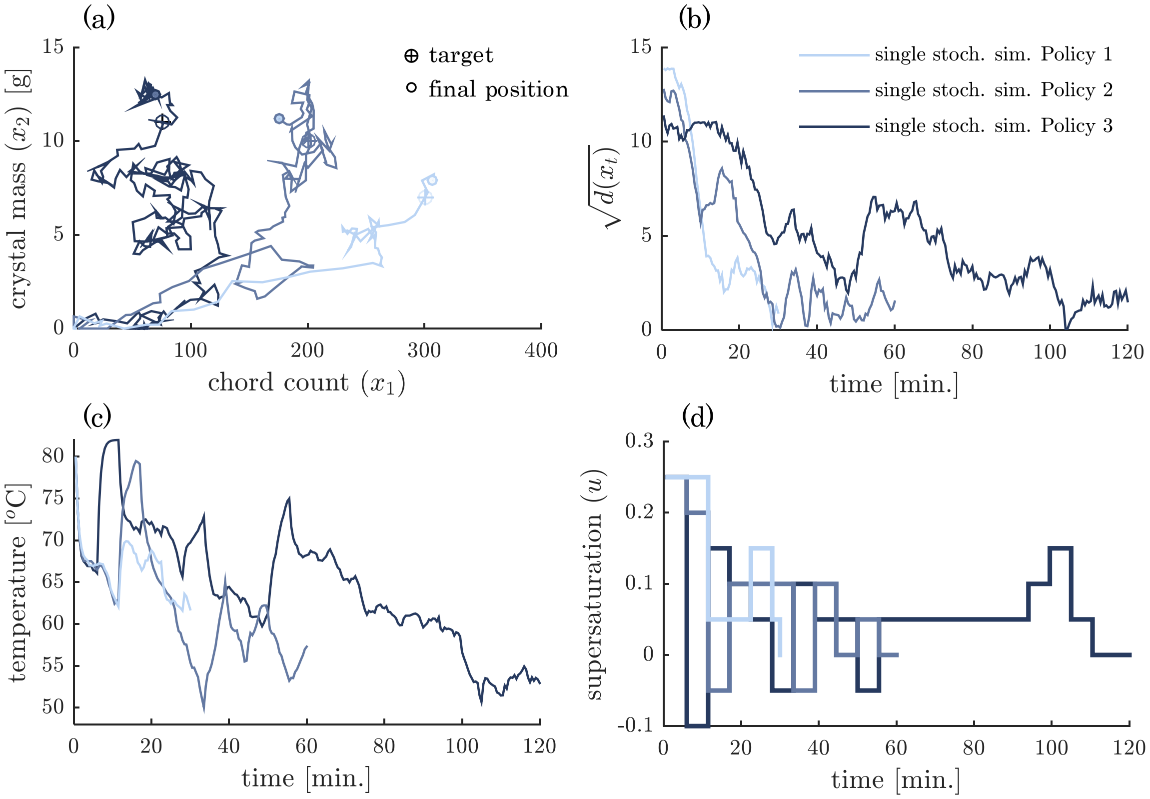

Figure C.7: Profiles for single stochastic simulations under each of the control policies: a.) crystallization trajectories; b.) distance-to-target profiles; c.) temperature profiles; d.) supersaturation setpoint profiles.

While the distance-to-target does not decay as smoothly as in the deterministic simulations, the policies do eventually move the mass-count position near the target position in each case—suggesting that the feedback policies are robust and work even with stochastic dynamics, measurement error, and actuation error. This is typical of the stochastic simulations [Griffin et al., 2016a].

As a whole, the stochastic simulation results suggest that the feedback control policies can be expected to drive the system toward the target, even when there is a significant random component to the dynamics, measurement error, and actuation delay.

APPENDIX D

PARAMETERS AND COMPUTATION: FEEDBACK CONTROL

D.1 sGATEC Control Parameters

Although the sGATEC scheme was constructed from a qualitative understanding of the effect of temperature manipulations on the movement in MC space, codification requires quantitative control parameters to be specified. A list of the control parameters is given in Table D.1 along with the values we have used. These parameters were selected considering the following constraints and heuristics:

- **The mass threshold**, m^0, should be selected such that a crystal mass measurement above this threshold clearly indicates the presence of crystals. We have set m^0 to 0.75g for darapskite.

- **The Heating Zone angles,** θ_{H1} and θ_{H2} must satisfy the following inequalities:

  $$0 \leq \theta_{\text{H1}} < \theta_{\text{H2}} \leq 90°.$$

  In addition, when selecting θ_{H2} we have to consider the target position in relation to the origin and satisfy the constraint:

  $$\theta_{\text{0-T}} < \theta_{\text{H2}},$$

  where $\theta_{\text{0-T}}$ is the angle between the origin and the target.

  To choose the Heating Zone angles from the set that satisfies the above constraints, we recognize that the angles define the width of the Heating-Deactivation

Zone. The angles should therefore be selected to create a substantial Heating-Deactivation Zone that is, at the same time, different from the Cooling Zone. If the Heating-Deactivation Zone is too small—i.e. $\theta_{\mathrm{H1}} \approx \theta_{\mathrm{H2}}$—we run the risk of remaining in Heating Mode indefinitely. On the other hand, if the Heating-Deactivation Zone is the same as the Cooling Zone—i.e. if θ_{H1} is set to 0°and θ_{H2} is set to 90°—we run the risk of excessive toggling back and forth between cooling and heating. We have set θ_{H1} to 10°and θ_{H2} to 60–80°depending on the target position.

- **The Target Zone size** should be selected considering the tradeoff between the desired crystal size accuracy and control feasibility. We have specified the Target Zone to be the MC area within ±10% of the selected target position.

- **The linear heating and cooling rates** should be achievable by the temperature control system. We have selected "moderate" rates that can be easily implemented using our experimental system: the heating rate is set to 1.0 °C/minute and the cooling rate set to −0.5 °C/min.

- **The supersaturation setpoint function**, $\sigma_{\mathrm{SP}} = f\left(\theta_{\text{P-T}}\right)$, should be selected such that higher supersaturation setpoints are implemented when the angle, $\theta_{\text{P-T}}$, towards the target is low. In this case, dynamic data at different supersaturation levels is available that relates the crystallization angle to the level of supersaturation—given in Figure D.1. A simple, discrete function was used to capture the relationship.

The control parameters for sGATEC applied to darapskite crystallization are given in Table **??**.

Table D.1: sGATEC control paramters.

Parameter	Set Value
mass threshold m^0	0.75g
Heating Zone Angle 1, θ_{H1}	10°
Heating Zone Angle 2, θ_{H2}	60–80°
Target Zone size	±10%
linear cooling rate	−0.5°C/min.
heating rate	1.0°C/min.
$\sigma_{\text{SP}} = f\left(\theta_{\text{P-T}}\right)$	$^* \ \sigma_{\text{SP}} = \left\{ \begin{array}{ll} 0.1 & \theta_{\text{P-T}} \geq 40^\circ \\ 0.2 & 40^\circ > \theta_{\text{P-T}} \geq 25^\circ \\ 0.3 & \text{else} \end{array} \right\}$

*based off data given in Figure D.1

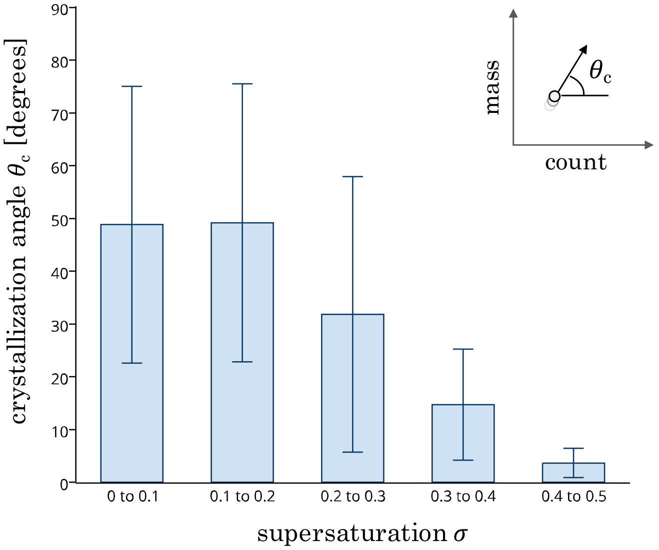

Figure D.1: Average crystallization angle at different levels of positive molar supersaturation. Error bars represent the standard deviation of the experimentally-observed crystallization angles.

D.2 Dynamic Programming

D.2.1 Discretization: dynamics as a cell-to-cell mapping

The optimization problem given by (6.3) can be approximately solved by discretizing the state space (MC space) and the input (supersaturation levels) and then applying dynamic programming.

Let $\mathcal{X}$ denote a set of center points for the cells of the discretized state space. And, let $\mathcal{U}$ denote a discrete set of inputs (supersaturation levels). F which previously described the dynamics as a point-to-point mapping, can now be converted to a cell-to-cell mapping [Bursal and Hu, 1989, Sun, 2013]:

$$\mathbf{x}_{\tau+1} = \mathcal{C}(\mathbf{x}_\tau, u_\tau, \Delta t) \equiv \underset{\mathbf{x}_{\tau+1} \in \mathcal{X}}{\text{argmin}} \quad ||\mathbf{x}_{\tau+1} - \big(F(\mathbf{x}_\tau, u_\tau)\Delta t + \mathbf{x}_\tau\big)||_2^2 ,$$

where $\mathcal{C}$ maps the current position in the grid, $\mathbf{x}_\tau \in \mathcal{X}$, under the input, $u_\tau \in \mathcal{U}$, to a new position in the grid, $\mathbf{x}_\tau \in \mathcal{X}$, over the time step Δt. This change in dynamic model is illustrated in Figure D.2.

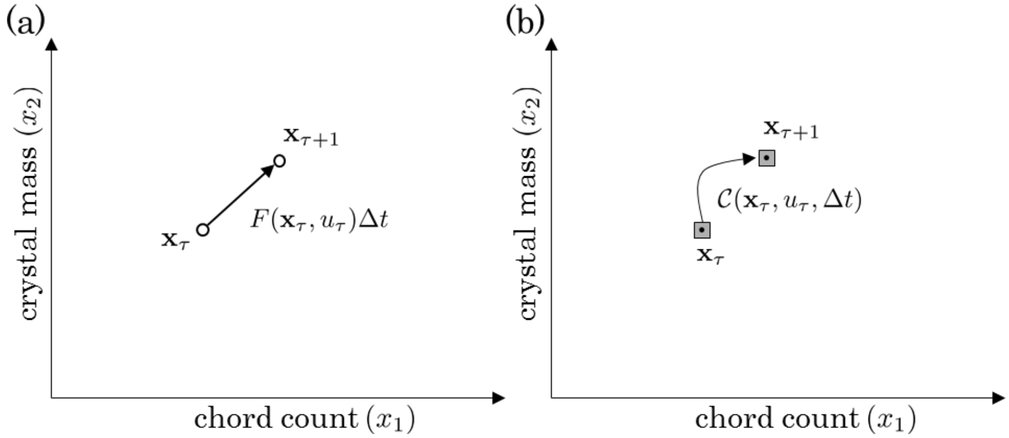

Figure D.2: Math model representations: (a) point-to-point dynamics; (b) cell-to-cell mapping.

D.2.2 Bellman or Dynamic Programming Principle

With the dynamics represented as a cell-to-cell mapping, we can apply the Bellman or Dynamic Programming Principle to obtain the optimal state-feedback control policy

[Bellman, 1952, Bellman, 1957]. This algorithm is outlined in Table D.2.

Table D.2: Dynamic programming algorithm for solving optimization problem (6.3).

```
Dynamic Programming Algorithm        Optimal Feedback Policy
set $V_N(\mathbf{x}_N) = d(\mathbf{x}_N, \mathbf{x}^{\oplus})$ for each $\mathbf{x}_N \in \mathcal{X}$
for $\tau = N-1, \ldots, 0$
    for each $\mathbf{x} \in \mathcal{X}$
$\phi^*_\tau(\mathbf{x}) \equiv u^*_\tau \mid_{\mathbf{x}} = \underset{u_\tau \in \mathcal{U}}{\operatorname{argmin}} \{(t_\tau/t_N)^\gamma d(\mathbf{x}, \mathbf{x}^{\oplus}) + \rho\varepsilon(u_\tau) + V_{\tau+1}(\mathbf{x}_{\tau+1})\}$
        subject to $\mathbf{x}_{\tau+1} = \mathcal{C}\big(\mathbf{x}, u_\tau, \Delta t\big)$;
    and
    $V_\tau(\mathbf{x}) = (t_\tau/t_N)^\gamma d(\mathbf{x}, \mathbf{x}^{\oplus}) + \rho\varepsilon\big(u^*_\tau \mid_{\mathbf{x}}\big) + V_{\tau+1}\Big(\mathcal{C}\big(\mathbf{x}, u^*_\tau \mid_{\mathbf{x}}, \Delta t\big)\Big)$.

combine to construct the optimal feedback control policy
    $\pi^* = \{\phi_0(\mathbf{x}), \ldots, \phi_{N-1}(\mathbf{x})\}$
```

Note: the optimization problem posed at each step and state can be solved by enumeration.

APPENDIX E

CODE FOR MACHINE LEARNING AND DYNAMIC PROGRAMMING

Code for Machine Learning (c.f. §5.2.1) and dynamic programming was developed in MATLAB. Some of this code has been made available online at:

http://dgriffin36.github.io/Crystallization-Control-Feedback-Policy-from-Data/.

Note: CVX, Software for Disciplined Convex Programming [Boyd and Vandenberghe, 2009, Grant and Boyd, 2012], is required to run the machine learning algorithm. This must be installed and in the appropriate path: http://cvxr.com/cvx/download/. For commercial use with non-free solvers, such as MATLAB, please obtain the appropriate license (http://cvxr.com/cvx/licensing/).

REFERENCES

[PAT, 2004] (2004). Guidance for Industry: PAT - A Framework for Innovative Pharmaceutical Development, Manufacturing, and Quality Assurance.

[Abbas and Romagnoli, 2006] Abbas, A. and Romagnoli, J. (2006). DCS Implementation of Optimal Operational Policies: A Crystallization Case Study. *International Journal of Computer Applications in Technology*, 25:198–208.

[Abu-Mostafa et al., 2012] Abu-Mostafa, Y. S., Hsuan-Tien, L., and Magdon-Ismail, M. (2012). *Learning From Data*. AML Book.

[Acevedo et al., 2015] Acevedo, D., Tandy, Y., and Nagy, Z. K. (2015). Multiobjective Optimization of an Unseeded Batch Cooling Crystallizer for Shape and Size Manipulation. *Industrial & Engineering Chemistry Research*, 54:2156–2166.

[Alessio and Bemporad, 2009] Alessio, A. and Bemporad, A. (2009). *Nonlinear Model Predictive Control: Towards New Challenging Applications*, chapter A Survey on Explicit Model Predictive Control, pages 345–369. Springer Berlin Heidelberg, Berlin, Heidelberg.

[Bakar et al., 2009a] Bakar, M. R. A., Nagy, Z. K., and Rielly, C. D. (2009a). Seeded Batch Cooling Crystallization with Temperature Cycling for the Control of Size Uniformity and Polymorphic Purity of Sulfathiazole Crystals. *Organic Process Research & Development*, 13:1343–1356.

[Bakar et al., 2009b] Bakar, M. R. A., Nagy, Z. K., Saleemi, A. N., and Rielly, C. D. (2009b). The Impact of Direct Nucleation Control on Crystal Size Distribution in Pharmaceutical Crystallization Processes. *Crystal Growth & Design*, 9:1378–1384.

[Barrett et al., 2010] Barrett, M., McNamara, M., Hao, H. X., Barrett, P., and Glennon, B. (2010). Supersaturation Tracking for the Development, Optimization and Control of Crystallization Processes. *Chemical Engineering Research and Design*, 88:1108–1119.

[Barrett and Glennon, 2002] Barrett, P. and Glennon, B. (2002). Characterizing the Metastable Zone Width and Solubility Curve Using Lasentec FBRM and PVM. *Chemical Engineering Research and Design*, 80:799–805.

[Barrett et al., 2005] Barrett, P., Smith, B., Worlitschek, J., Bracken, V., O'Sullivan, B., and O'Grady, D. (2005). Review of the Use of Process Analytical Technology for the Understanding and Optimization of Production Batch Crystallization Processes. *Advanced Drug Delivery Reviews*, 56:349–369.

[Baumgartner et al., 2014] Baumgartner, J., Dey, A., Bomans, P. H. H., Coadou, C. L., Fratzl, P., Sommerdijk, N. A. J. M., and Faivre, D. (2014). Nucleation and Growth of Magnetite from Solution. *Nature Materials*, 12:310–314.

[Bellman, 1952] Bellman, R. E. (1952). On the Theory of Dynamic Programming. *Proceedings of the National Academy of Sciences*, 38:716–719.

[Bellman, 1957] Bellman, R. E. (1957). *Dynamic Programming*. Princeton University Press.

[Bird et al., 2007] Bird, R. B., Stewart, W. E., and Lightfoot, E. N. (2007). *Transport Phenomena*. John Wiley & Sons, Inc., 2 edition.

[Boyd and Vandenberghe, 2009] Boyd, S. and Vandenberghe, L. (2009). *Convex Optimization*. Cambridge University Press.

[Braatz, 2002] Braatz, R. D. (2002). Advanced Control of Crystallization Processes. *International Journal of Modern Physics B*, 16:346–353.

[Braatz et al., 2002] Braatz, R. D., Fujiwara, M., Ma, D. L., Togkalidou, T., and Tafti, D. K. (2002). Simulation and New Sensor Technologies for Industrial Crystallization: A Review. *Annual Reviews in Control*, 26:87–99.

[Bursal and Hu, 1989] Bursal, H. F. and Hu, C. S. (1989). Application of a Cell-Mapping Method to Optimal-Control Problems. *International Journal of Control*, 49:1505–1522.

[Chang and Epstein, 1987] Chang, C. T. and Epstein, M. A. F. (1987). Simulation Studies of a Feedback Control Strategy for Batch Crystallizers. *AIChE Symposium Series*, 83:110–119.

[Chew et al., 2007] Chew, J. W., Chow, P. S., and Tan, R. B. H. (2007). Automated In-line Technique Using FBRM to Achieve Consistent Product Quality in Cooling Crystallization. *Crystal Growth & Design*, 7:1416–1422.

[Cornel et al., 2008] Cornel, J., Lindenberg, C., and Mazzotti, M. (2008). Quantitative Application of In Situ ATR-FTIR and Raman Spectroscopy in Crystallization Processes. *Industrial & Engineering Chemistry Research*, 47:4870–4882.

[Corriou and Rohani, 2008] Corriou, J.-P. and Rohani, S. (2008). A New Look at Optimal Control of a Batch Crystallizer. *Aiche Journal*, 54:3188–3206.

[Derdour et al., 2003] Derdour, L., Fevotte, G., Puel, F., and Carvin, P. (2003). Real-Time Evaluation of the Concentration of Impurities during Organic Solution Crystallization. *Powder Technology*, 129:1–7.

[Doki et al., 2004] Doki, N., Seki, H., Takano, K., Asatani, H., Yokota, M., and Kubota, N. (2004). Process Control of Seeded Batch Cooling Crystallization of the Metastable α-Form Glycine Using an In-Situ ATR-FTIR Spectrometer and an In-Situ FBRM Particle Counter. *Crystal Growth & Design*, 4:949–953.

[Dunuwila and Berglund, 1997] Dunuwila, D. D. and Berglund, K. A. (1997). ATR-FTIR Spectroscopy for In Situ Measurement of Supersaturation. *Journal of Crystal Growth*, 179:185–193.

[Dunuwila et al., 1994] Dunuwila, D. D., Carroll, L. B., and Berglund, K. A. (1994). An Investigation of the Applicability of Attenuated Total-Reflection Infrared Spectroscopy for Measurement of Solubility and Supersaturation of Aqueous Citric-Acid Solutions. *Journal of Crystal Growth*, 137:561–568.

[Erdemir et al., 2009] Erdemir, D., Lee, A. Y., and Myerson, A. S. (2009). Nucleation of Crystals from Solution: Classical and Two-Step Models. *Accounts of Chemical Research*, 44(5):621–629.

[Erdemir et al., 2013] Erdemir, D., Lee, A. Y., and Myerson, A. S. (2013). Nucleation: More than one pathway. *Nature Materials*, 12:284–285.

[Feng and Berglund, 2002] Feng, L. L. and Berglund, K. A. (2002). ATR-FTIR for Determining Optimal Cooling Curves for Batch Crystallization of Succinic Acid. *Crystal Growth & Design*, 2:449–452.

[Fujiwara et al., 2002] Fujiwara, M., Chow, P. S., Ma, D. L., and Braatz, R. D. (2002). Paracetamol Crystallization Using Laser Backscattering and ATR-FTIR Spectroscopy: Metastability, Agglomeration, and Control. *Crystal Growth & Design*, 2:363–370.

[Geladi and Kowalski, 1986] Geladi, P. and Kowalski, B. R. (1986). Partial Least-Squares Regression—a Tutorial. *Analytica Chimica Acta*, 185:1–17.

[Gibbs, 1876] Gibbs, J. W. (1876). On the Equilibrium of Hetergeneous Substances. *Transactions of the Connecticut Academy of Arts and Sciences*, 3:108–248.

[Gibbs, 1878] Gibbs, J. W. (1878). On the Equilibrium of Hetergeneous Substances. *Transactions of the Connecticut Academy of Arts and Sciences*, 16:343–524.

[Grant and Boyd, 2012] Grant, M. and Boyd, S. (2012). CVX: MATLAB Software for Disciplined Convex Programming. http://cvxr.com/cvx.

[Griffin et al., 2014] Griffin, D. J., Grover, M. A., Kawajiri, Y., and Rousseau, R. W. (2014). Robust Multicomponent IR-to-Concentration Model Regression. *Chemical Engineering Science*, 116:77–99.

[Griffin et al., 2015a] Griffin, D. J., Grover, M. A., Kawajiri, Y., and Rousseau, R. W. (2015a). Combining ATR-FTIR and FBRM for Feedback on Crystal Size. In *American Control Conference (ACC)*. Chicago, IL.

[Griffin et al., 2015b] Griffin, D. J., Grover, M. A., Kawajiri, Y., and Rousseau, R. W. (2015b). Controlled Crystallization of Salts from Nuclear Waste Solutions. In *Waste Management (WM) Conference*. Phoenix, AZ.

[Griffin et al., 2015c] Griffin, D. J., Grover, M. A., Kawajiri, Y., and Rousseau, R. W. (2015c). Mass-Count Plots for Crystal Size Control. *Chemical Engineering Science*, 137:338–351.

[Griffin et al., 2016a] Griffin, D. J., Grover, M. A., Kawajiri, Y., and Rousseau, R. W. (2016a). Data-Driven Modeling and Dynamic Programming Applied to Batch Cooling Crystallization. *Industrial & Engineering Chemistry Research*, 55:1361–1372.

[Griffin et al., 2015d] Griffin, D. J., Kawajiri, Y., Grover, M. A., and Rousseau, R. W. (2015d). Feedback Control of Multicomponent Salt Crystallization. *Crystal Growth & Design*, 15:305–317.

[Griffin et al., 2016b] Griffin, D. J., Tang, X., and Grover, M. A. (2016b). Externally-Directing Self-Assembly with Dynamic Programming. In *American Control Conference (ACC)*. Submitted.

[Griffiths, 1924] Griffiths, H. (1924). Mechanical Crystallization. *Journal of the Chemical Society, Transactions*, 44:7T–18T.

[Gron et al., 2002] Gron, H., Borissova, A., and Roberts, K. J. (2002). In-Process ATR-FTIR Spectroscopy for Closed-Loop Supersaturation Control of a Batch Crystallizer Producing Monosodium Glutamate Crystals of Defined Size. *Industrial & Engineering Chemistry Research*, 42:198–206.

[Heinrich and Ulrich, 2012] Heinrich, J. and Ulrich, J. (2012). Application of Laser-Backscattering Instruments for In Situ Monitoring of Crystallization Processes—A Review. *Chemical Engineering & Technology*, 35:967–979.

[Herting, 1996] Herting, D. L. (1996). Clean Salt Process—Final Report. Technical report, Department of Energy, Environmental Management.

[Hu et al., 2005] Hu, Q., Rohani, S., Wang, D. X., and Jutan, A. (2005). Optimal Control of a Batch Cooling Seeded Crystallizer. *Powder Technology*, 156:170–176.

[Hulburt and Katz, 1964] Hulburt, H. M. and Katz, S. (1964). Some Problems in Particle Technology—A Statistical Mechanical Formulation. *Chemical Engineering Science*, 19:555–574.

[Jones and Mullin, 1974] Jones, A. G. and Mullin, J. W. (1974). Programmed Cooling Crystallization of Potassium Sulphate Solutions. *Chemical Engineering Science*, 29:105–118.

[Kee et al., 2011] Kee, N. C. S., Tan, R. B. H., and Braatz, R. D. (2011). Semi-automated Identification of the Phase Diagram for Enantiotropic Crystallizations Using ATR-FTIR Spectroscopy and Laser Backscattering. *Industrial & Engineering Chemistry Research*, 50:1488–1495.

[Lang et al., 1999] Lang, Y. D., Cervantes, A. M., and Biegler, L. T. (1999). Dynamic Optimization of a Batch Cooling Crystallization Process. *Industrial & Engineering Chemistry Research*, 38:1469–1477.

[Larsen et al., 2006] Larsen, P. A., Patience, D. B., and Rawlings, J. B. (2006). Industrial Crystallization Process Control. *Ieee Control Systems Magazine*, 26:70–80.

[Lewiner et al., 2001] Lewiner, F., Klein, J. P., Puel, F., and Fevotte, G. (2001). Online ATR-FTIR Measurement of Supersaturation During Solution Crystallization Processes. Calibration and Applications on Three Solute/Solvent Systems. *Chemical Engineering Science*, 56:2069–2084.

[Lewis et al., 2015] Lewis, A. E., Seckler, M. M., Kramer, H., and van Rosmalen, G. (2015). *Industrial Crystallization: Fundamentals and Applications*. Cambridge University Press.

[Li et al., 2013] Li, H., Grover, M. A., Kawajiri, Y., and Rousseau, R. W. (2013). Development of an Empirical Method Relating Crystal Size Distribution and FBRM Measurements. *Chemical Engineering Science*, 89:142–151.

[Lindenberg et al., 2009] Lindenberg, C., Krattli, M., Cornel, J., and Mazzotti, M. (2009). Design and Optimization of a Combined Cooling/Antisolvent Crystallization Process. *Crystal Growth & Design*, 9:1124–1136.

[Liotta and Sabesan, 2004] Liotta, V. and Sabesan, V. (2004). Monitoring and Feedback Control of Supersaturation using ATR-FTIR to Produce an Active Pharmaceutical Ingredient of a Desired Crystal Size. *Organic Process Research & Development*, 8:488–494.

[Matthews and Rawlings, 1998] Matthews, H. B. and Rawlings, J. B. (1998). Batch Crystallization of a Photochemical: Modeling, Control, and Filtration. *AIChE Journal*, 44:1119–1127.

[McCabe, 1929] McCabe, W. L. (1929). Crystal Growth in Aqueous Solutions. *Industrial & Engineering Chemistry*, 21:112–119.

[Mesbah et al., 2011] Mesbah, A., Huesman, A. E. M., Kramer, H. J. M., Nagy, Z. K., and den Hof, P. M. J. V. (2011). Real-Time Control of a Semi-Industrial Fed-Batch Evaporative Crystallizer Using Different Direct Optimization Strategies. *AIChE Journal*, 57:1557–1569.

[Mesbah et al., 2010] Mesbah, A., Landlust, J., Huesman, A. E. M., Kramer, H. J. M., Jansens, P. J., and den Hof, P. M. J. V. (2010). A model-based control framework for industrial batch crystallization processes. *Chemical Engineering Research & Design*, 88:1188–1201.

[Mesbah et al., 2012] Mesbah, A., Nagy, Z. K., Huesman, A. E. M., Kramer, H. J. M., and den Hof, P. M. J. V. (2012). Nonlinear Model-Based Control of a Semi-Industrial Batch Crystallizer Using a Population Balance Modeling Framework. *Ieee Transactions on Control Systems Technology*, 20:1223–1233.

[Miller and Rawlings, 1994] Miller, S. M. and Rawlings, J. B. (1994). Model Identification and Control Strategies for Batch Cooling Crystallizers. *Aiche Journal*, 40:1312–1327.

[Montillon and Badger, 1927] Montillon, G. and Badger, W. L. (1927). Rate of Growth of Crystals in Aqueous Solution. *Industrial & Engineering Chemistry*, 19:809–816.

[Mullin and Nyvlt, 1971] Mullin, J. and Nyvlt, J. (1971). Programmed Cooling of Batch Crystallizers. *Chemical Engineering Science*, 26:369–377.

[Mullin, 2001] Mullin, J. W. (2001). *Crystallization.* Elsevier Butterworth-Heinemann, 4 edition.

[Nagy, 2009] Nagy, Z. K. (2009). Model Based Robust Control Approach for Batch Crystallization Product Design. *Computers & Chemical Engineering*, 33:1685–1691.

[Nagy and Braatz, 2004] Nagy, Z. K. and Braatz, R. D. (2004). Open-loop and Closed-loop Robust Optimal Control of Batch Processes using Distributional and Worst-case Analysis. *Journal of Process Control*, 14:411–422.

[Nagy and Braatz, 2012] Nagy, Z. K. and Braatz, R. D. (2012). Advances and New Directions in Crystallization Control. *Annual Review of Chemical and Biomolecular Engineering*, 3:55–75.

[Nagy et al., 2013] Nagy, Z. K., Fevotte, G., Kramerand, H., and Simon, L. L. (2013). Recent Advances in the Monitoring, Modeling and Control of Crystallization Systems. *Chemical Engineering Research & Design*, 91:1903–1922.

[Nassif et al., 2008] Nassif, L., Dumont, G., Alysouri, H., and Rousseau, R. W. (2008). Pretreatment of Hanford Medium-Curie Wastes by Fractional Crystallization. *Environmental Science &Technology*, 42:4940–4945.

[Nielsen et al., 2014] Nielsen, M. H., Aloni, S., and Yoreo, J. J. D. (2014). In Situ TEM Imaging of CaCO3 Nucleation Reveals Coexistence of Direct and Indirect Pathways. *Science*, 345:1158–1162.

[Ramkrishna, 2000] Ramkrishna, D. (2000). *Population Balances: Theory and Applications to Particulate Systems in Engineering.* Elsevier Inc.

[Randolph, 1964] Randolph, A. D. (1964). Population Balance For Countable Entities. *Canadian Journal of Chemical Engineering*, 42(6):280–281.

[Randolph and Larson, 1988] Randolph, A. D. and Larson, M. A. (1988). *Theory of Particulate Processes*. San Diego: Academic Press, 2 edition.

[Rawlings et al., 1993] Rawlings, J. B., Miller, S. M., and Witkowski, W. R. (1993). Model Identification and Control of Solution Crystallization Processes—A Review. *Industrial & Engineering Chemistry Research*, 32:1275–1296.

[Saleemi et al., 2012] Saleemi, A., Rielly, C., and Nagy, Z. K. (2012). Automated Direct Nucleation Control for In Situ Dynamic Fines Removal in Batch Cooling Crystallization. *CrystEngComm*, 14:2196–2203.

[Sheikhzadeh et al., 2007] Sheikhzadeh, M., Trifkovic, M., and Rohani, S. (2007). Real-Time Opimal Control of an Anti-Solvent Isothermal Semi-Batch Crystallizer. *Chemical Engineering Science*, 63:S562–S574.

[Simon et al., 2015] Simon, L. L., Pataki, H., Marosi, G., Meemken, F., Hungerbuhler, K., Baiker, A., Tummala, A. S., Glennon, B., Kuentz, M., Steele, G., Kramer, H. J. M., Rydzak, J. W., Chen, Z., Morris, J., Kjell, F., Singh, R., Gani, R., Gernaey, K. V., Louhi-Kultanen, M., O'Reilly, J., Sandler, N., Antikainen, O., Yliruusi, J., Frohberg, P., Ulrich, J., Braatz, R. D., Leyssens, T., von Stosch, M., Oliveira, R., Tan, R. B. H., Wu, H., Khan, M., O'Grady, D., Pandey, A., Westra, R., Delle-Case, E., Pape, D., Angelosante, D., Maret, Y., Steiger, O., Lenner, M., Abbou-Oucherif, K., Nagy, Z. K., Litster, J. D., Kamaraju, V. K., and Chiuq, M. (2015). Assessment of Recent Process Analytical Technology (PAT) Trends: A Multiauthor Review. *Organic Process Research & Development*, 19:3–62.

[Smola and Scholkopf, 2004] Smola, A. J. and Scholkopf, B. (2004). A Tutorial on Support Vector Regression. *Statistics and Computing*, 14:199–222.

[Sun, 2013] Sun, J. Q. (2013). Control of Nonlinear Dynamic Systems with the Cell Mapping Method. *Advances in Intelligent Systems and Computing*, 175:3–18.

[Togkalidou et al., 2001] Togkalidou, T., Fujiwara, M., Patel, M., and Braatz, R. D. (2001). Solute Concentration Prediction using Chemometrics and ATR-FTIR Spectroscopy. *Journal of Crystal Growth*, 231:534–543.

[Togkalidou et al., 2002] Togkalidou, T., Tung, H. H., Sun, Y. K., Andrews, A., and Braatz, R. D. (2002). Solution Concentration Prediction for Pharmaceutical Crystallization Processes using Robust Chemometrics and ATR-FTIR Spectroscopy. *Organic Process Research & Development*, 6:317–322.

[Vekilov, 2010] Vekilov, P. G. (2010). Nucleation. *Crystal Growth & Design*, 10:5007–5019.

[White and Write, 1971] White, E. T. and Write, P. G. (1971). Magnitude of Size Dispersion Effects in Crystallization. *Chemical Engineering Progress Symposium Series*, 67:81.

[Wold et al., 1984] Wold, S., Ruhe, A., Wold, H., and Dunn, W. J. (1984). The Collinearity Problem in Linear Regression—the Partial Least-Squares (PLS) Approach to Generalized Inverses. *SIAM Journal on Scientific and Statistical Computing*, 5:735–743.

[Woo et al., 2009] Woo, X. Y., Nagy, Z. K., Tan, R. B. H., and Braatz, R. D. (2009). Adaptive Concentration Control of Cooling and Antisolvent Crystallization with Laser Backscattering Measurement. *Crystal Growth & Design*, 9:182–191.

[Worlitschek and Mazzotti, 2004] Worlitschek, J. and Mazzotti, M. (2004). Model-Based Optimization of Particle Size Distribution in Batch-Cooling Crystallization of Paracetamol. *Crystal Growth & Design*, 4:891–903.

[Xue et al., 2014] Xue, Y., Beltran-Villegas, D. J., Tang, X., Bevan, M. A., and Grover, M. A. (2014). Optimal Design of a Colloidal Self-Assembly Process. *IEEE TRANSACTIONS ON CONTROL SYSTEMS TECHNOLOGY*, 22:1956–1963.

[Yang and Nagy, 2015] Yang, Y. and Nagy, Z. K. (2015). Advanced Control Approaches for Combined Cooling/Antisolvent Crystallization in Continuous Mixed Suspension Mixed Product Removal Cascade Crystallizers. *Chemical Engineering Science*, 127:362–373.

[Yu et al., 2004] Yu, L. X., Lionberger, R. A., Raw, A. S., D'Costa, R., Wu, H. Q., and Hussain, A. S. (2004). Applications of Process Analytical Technology to Crystallization Processes. *Organic Process Research & Development*, 9:348–355.

[Yu et al., 2006] Yu, Z. Q., Chow, P. S., and Tan, R. B. H. (2006). Seeding and Constant-Supersaturation Control by ATR-FTIR in Anti-Solvent Crystallization. *Organic Process Research & Development*, 10:717–722.

[Zhang and Rohani, 2003] Zhang, G. P. and Rohani, S. (2003). On-line Optimal Control of a Seeded Batch Cooling Crystallizer. *Chemical Engineering Science*, 58:1887–1896.

[Zhou et al., 2006] Zhou, V. X., Fujiwara, M., Woo, X. Y., Rusli, E., Tung, H. H., Starbuck, C., Davidson, O., Ge, Z., and Braatz, R. D. (2006). Direct Design of Pharmaceutical Antisolvent Crystallization Through Concentration Control. *Crystal Growth & Design*, 6:892–898.

[Zumstein and Rousseau, 1987a] Zumstein, R. C. and Rousseau, R. W. (1987a). Growth Rate Dispersion by Initial Growth Rate Distributions and Growth Rate Fluctuations. *AIChE Journal*, 33:121–129.

[Zumstein and Rousseau, 1987b] Zumstein, R. C. and Rousseau, R. W. (1987b). Growth Rate Dispersion in Batch Crystallization with Transient Conditions. *AIChE Journal*, 33:1921–1925.