

HYBRID MODELING AND ROBUSTNESS ANALYSIS OF CELL CYCLE REGULATION

A DISSERTATION

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL ENGINEERING

AND THE COMMITTEE ON GRADUATE STUDIES

OF STANFORD UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

Xiling Shen

August 2008

© Copyright by Xiling Shen 2008
All Rights Reserved

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Mark A. Horowitz) Principal Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(Harley H. McAdams)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

(David L. Dill)

Approved for the University Committee on Graduate Studies

Abstract

Caulobacter crescentus is a model organism for studying asymmetrical bacteria cell cycle division. During cell cycle, a *Caulobacter* cell needs to accomplish processive molecular functions in the right sequence: It sheds its flagella, grows a stalk, replicates and segregates its chromosomes, and initiates cytokinesis to compartmentalize the two morphologically distinct daughter cells, all of which are coordinated by a genetic control circuit comprised of cascaded regulatory proteins that are expressed in an orderly and timely fashion. Non-genetic mechanisms like methylation-based promoter control, phospho-signal pathway and regulated proteolysis couples the succession of regulator expression back to the progression of various cell cycle processes, closing various feedback control loops. With advances in experimental technology, the understanding of this cellular regulatory system has improved to the level of complexity which is difficult for intuitive understanding, especially when dynamic behaviors resulting from feedback effects are concerned.

To simulate the dynamic properties of the cell cycle feedback control, an *in silico* simulation model is constructed based on the concept of hybrid system from control theory. Mimicking the *Caulobacter* control structure *in vivo*, the hybrid model uses continuous ordinary differential equations (ODE) to model well-understood molecular reactions such as protein synthesis, but uses discrete event-driven finite state machines (FSM) to phenomenologically model complex cell processes and instantaneous reactions. Hybrid models create a flexible and extensible architecture capable of handling different levels of abstraction and lack of complete knowledge, the two biggest challenges of modeling biological systems. The model was validated by the consistency between simulation results and experimental measurements including protein and

mRNA concentration profiles. *In silico* mutants based on our model further managed to correctly predict phenotypes of various *in vivo* mutants.

Because most biological systems have to survive under a variety of environmental, genetic, and stochastic perturbations, it has been postulated that their regulatory systems have to be quite robust, so a further analysis of the robustness property of the modeled cell cycle regulation can provide new biological insights. Unfortunately it is difficult for traditional methods like parameter sensitivity analysis to fully explore the design space and intuitively interpret the result when facing a complex model. By creating an equivalent asynchronous digital circuit representation of the cell cycle model, which maintains properties of interest, we were able to apply formal model checking techniques to exhaustively search the entire state space to identify the potential scenarios which causes the cell cycle to fail to complete. The analysis revealed that the top level control of the *Caulobacter* cell cycle regulation is extremely robust with very few cases of potential failures; furthermore, non-genetic mechanisms such as methylation based control of promoter activation and its remaining basal expression have been shown to play an important role for robustness under special circumstances. Model checking also verified that the modeled cell cycle is able to robustly switch into growth arrest when facing stress or starvation.

Acknowledgments

My PhD experience at Stanford has been an amazing voyage, culminating in the work presented in this dissertation, which would not have been possible without the support of many incredible individuals with whom I had the fortune to work and collaborate.

First and foremost, I want to thank my principle advisor, Professor Mark Horowitz, whose mentorship has been very intellectually stimulating. Mark has a knack of bringing out student's full potential by encouraging them to explore creative and interdisciplinary ideas. He also painstakingly helped me work on fundamental research skills such as effective communication through writing and presentation as well as dissection of complex problems. Mark is a role model of an inspiring advisor who I would always strive to be in my future academic career.

I also want to thank Professor Harley McAdams who welcomed me into his lab when I expressed interest in doing biology related research and displayed incredible amount of patience with me, being methodical in teaching me the fundamental concepts and research approaches in biology. Harley's guidance has been indispensable for my conversion to the field of biology without any prior background.

I am indebted to Professor David Dill for working closely with me and taught me how to adapt model checking to biological systems. Dave has been instrumental in applying engineering techniques to problems in biology, and I have learned a great deal from him beyond the scope of this dissertation work.

My access to all the experimental data in Professor Lucy Shapiro's lab provides the backbone of the modeling work in this dissertation. Besides being a world-renowned researcher in her field, Lucy exudes warmth and joy, and together with Harley, run the entire lab as a family, of which I was fortunate enough to be a part.

There are many other faculties at Stanford who played pivotal roles in my PhD experience. I want to thank Professor Stephen Boyd for having first inspired my passion in research during my undergraduate years at Stanford and eventually serving as the chair of my oral committee. I want to thank Professor Joseph Khan for having worked with me for more than a year on a fascinating multi-mode fiber project. I want to thank Professor Claire Tomlin who first mentioned the concept of hybrid system to me. I want to thank Professor Markus Covert, an expert in systems biology, who gave me a lot of good advice. I want to thank Professor Douglas Brutlag who has been very supportive of my work. I also want to thank Professor Jim Ferrell with whom I had several good discussions.

The current and former students in both the Horowitz lab and the Shapiro & McAdams lab have contributed greatly to my work and life at Stanford. The first person I would like to thank is Dr. Justine Collier, who collaborated with me on many of the projects and provided most of the experimental support. I am also indebted to Valentin Abramzon, Elad Alon, Fernando Amat, Amir Amirkhany, Frances Lau, Hae-Chang Lee, Ken Mai, Bitá Neẓamfar, Sam Palermo, Dinesh Patil, Alex Solomatnikov, Vladimir Stojanović, and Jim Weaver from the Horowitz lab, Eduardo Abeliuk, Grant Bowman, Leticia Britos, Joseph Chen, Natalie Dye, Mike Fero, Erin Goley, Nathan Hillson, Sun-Hae Hong, Antonio Iniesta, Virginia Kalogeraki, Steve Landt, Jay Lesley, Sean Murray, Mohammed Quraishi, Monica Schwartz, Meng How Tang, Esteban Toro, Yi-Chun Yeh, Lisandra West, Ling Xie, and Balaji Srinivasan from the Shapiro & McAdams lab, and Debashis Sahoo from the Dill lab for their friendship including many delightful discussions and collaborations.

I would like to thank Teresa Lynn and Tara Trim for the wonderful administrative work they have done, which helped my research run smoothly.

I wish to thank MARCO C2S2, Texas Instruments (TI) and Department of Energy (DOE) for enabling my research with their financial support. I also want to thank my colleagues from Barcelona Design and Texas Instruments where I worked prior to and during the summers of my PhD. Special thanks to Dr. Mar Hershensen and Dr. Don Shavor for their continuing advice and support throughout my journey to PhD.

Lastly, my gratitude towards my family for their constant love and unwavering support is beyond words. My wife Bethany has always been behind me and took care of a disproportionate share of family responsibilities for me to focus on my research. My son Aidan has also endured

many hardships due to my student life for which I would never be able to compensate. I can never thank my parents and grandparents enough for the constant sacrifices they have made on my behalf. My mother Annie and my father Qi have always been my emotional support. Only owing to the unwavering confidence they and the rest of the family had put in me did I have the strength and courage to pursue my passion and arrive at this day.

Table of Contents

Abstract	v
Acknowledgments	vii
Table of Contents	x
Introduction	1
1.1 Mathematical modeling of biological systems	2
1.2 Analysis of robustness	4
1.3 <i>Caulobacter</i> cell cycle regulation	5
Caulobacter cell cycle regulation	7
2.1 cell cycle stages	7
2.2 Master regulatory network.....	9
2.3 Feedback mechanisms	11
2.4 system model	16
Chapter 3 Hybrid system modeling	19
3.1 Ordinary differential equation (ODE).....	21
3.2 Hybrid system	23

3.3 Model implementation in Matlab	26
Chapter 4 Simulation results	36
4.1 Validation.....	37
4.2 <i>In silico</i> mutant strain	43
Chapter 5 Robustness.....	46
5.1 Robustness analysis on system model	47
5.2 Discrete abstraction for model checking	49
5.3 Discrete representation of <i>Caulobacter</i> cell cycle regulation.....	53
5.4 Timing verification	56
5.5 Model checking results	6256
Chapter 6 Summary and discussion.....	67
6.1 Conclusion	67
6.2 Discussion and future work	69
Appendix A Regulator Models and Parameters	73
A.1 ODE models of master regulators.....	73
A.2 Parameters.....	79
Appendix B Simulated mRNA levels vs Microarray measurements	87
Appendix C <i>In silico</i> mutant simulation	91
Appendix D Implementation in NuSMV	103
Bibliography	107

Chapter 1

Introduction

With advances in experimental technology, the understanding of cellular regulatory systems has improved to the level of complexity which is difficult for intuitive understanding, especially when dynamic behaviors resulting from feedback effects are concerned{Lauffenburger, 2000 #829; McAdams, 1995 #787}. Such complexity hampers the exploration of further system level questions such as the underlying principles of these regulatory systems that provides evolutionary advantages. Mathematical modeling has long been used in engineering and other disciplines of science to synthesize existing information to describe a complex system through simulation. Facing the unique challenge in studying biological systems when complete knowledge is often not available, a successful modeling approach should be capable of incorporating both qualitative and quantitative descriptions into the complete description of a regulatory system by using different levels of abstraction. The abstraction also help engineering analysis from control theory and circuit design to reveal inconsistency and missing pieces in the knowledge base as well as provide clues to new experimental direction.

In this work, we present a system model of *Caulobacter* cell cycle regulation which copes with different levels of abstraction by using the concept of hybrid system from control theory. We then demonstrate how we adapted a formal verification technique called symbolic model checking from asynchronous digital circuit design to examine the robustness of the cell cycle model, which provides insights into the design of the control circuit.

1.1 Mathematical modeling of biological systems

Research in molecular biology typically provides a logic description of the inter-connections between the various genes and modules in a biological regulatory system, such as protein A activates or represses the expression of gene b. This level of description is often sufficient to explain the function of a stand alone pathway, but soon becomes overwhelming and non-intuitive for humans to follow when knowledge about the system accumulates and the description becomes more complex (Figure 1.1). Boolean networks {Huang, 1999 #833; Kauffman, 1969 #834; Thomas, 2001 #819; Davidson, 2002 #835} have been proposed to simulate regulatory networks based on logic descriptions, but it falls short again when the modeled system contains feedback mechanisms because the behavior of such a system now depends on the dynamic characteristics of these mechanisms in addition to its static topology, which is beyond the scope of a conventional biological description.

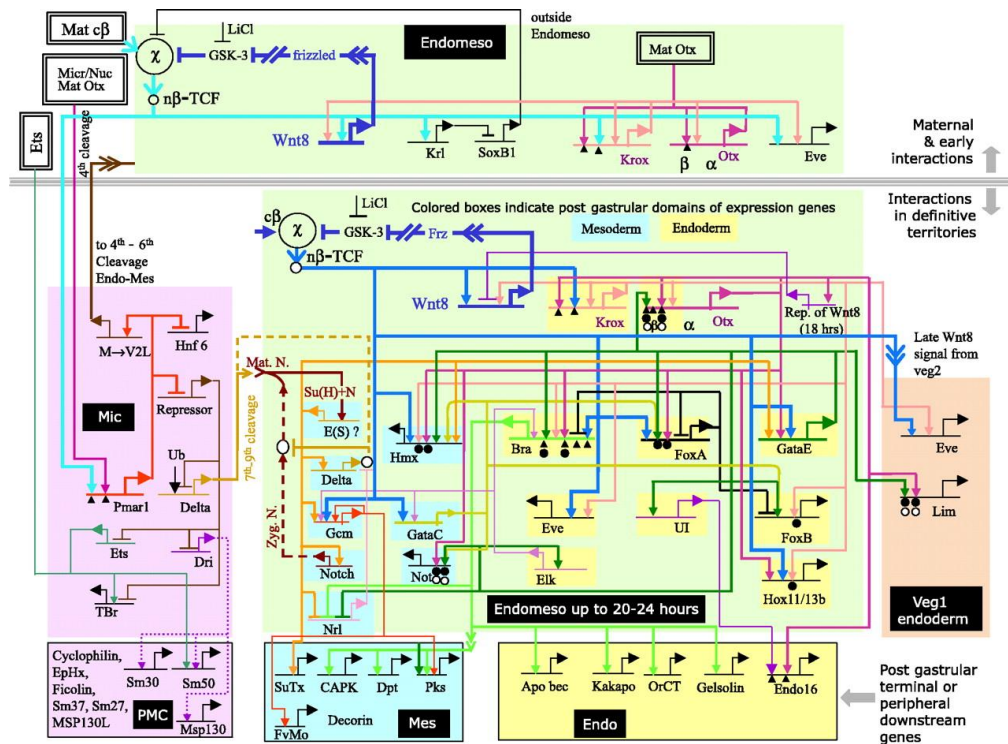


Figure 1.1 A regulatory network for sea urchin embryo development {Davidson, 2002 #835}

The idea of building computational models of biological systems to capture their dynamic behavior have been around for ages {Turing, 1990 #836}, but practical concerns such as the

complexity of biological systems and the lack of means to characterize them has limited its success until recently. Advances in theoretical and experimental methods dealing with complexity {Weng, 1999 #837} and quantitative modeling has given rise to the emergence of a new discipline: systems biology {Ideker, 2001 #830}, which tries to evaluate roles of individual pathways and their interactions in the context of the overall system behavior by constructing dynamic mathematical models of systems and sub-systems {Ideker, 2001 #830; Kitano, 2002 #831; Kitano, 2002 #832}. After validation these system models can then be used to make predictions on phenotypes and behaviors that are yet uncharacterized. The benefits of such models have encouraged efforts to construct whole cell simulation, in which an entire single cell organism is included in the model as a whole to understand system-wide functions {Schaff, 1999 #838; Tomita, 1999 #839}. The advent of high-throughput measurements like the Microarray technology provides a convenient way to validate the gene expression profile predicted by the model as well as allow various bioinformatics tools including clustering, motif finding {Milo, 2002 #840} and Bayesian network {Friedman, 2000 #841} to suggest abstract models to fill in the gaps of current understandings of these biological systems.

System models are traditionally constructed from the bottom-up by assembling experimental characterizations of individual pathways. This approach works well for smaller scale models of sub-systems, but in a more complex system, pathways not yet characterized by experiments and thus absent from the models often prevents holistic simulations from generating correct dynamic behaviors. The lack of levels of hierarchy and abstraction also makes models constructed from the bottom-up unintuitive and unsuitable for higher level analysis tools {Noble, 2003 #842}. In contrary, engineering models are regularly built from the top-down, in which abstract models of the complete system are progressively replaced by more detailed models of components parts. In the top-down approach, uncharacterized mechanism could remain as abstract models until more data are available, so the incompleteness of system knowledge does not become an inhibitor for model building. Abstraction hierarchy naturally introduced by the top-down approach also makes the model intuitive and extendable. To support this top-down approach and provide multiple levels of abstraction, hybrid systems {Alur, 2002 #790; Ghosh, 2004 #844; Mishra, 2003 #845; Schaff, 1999 #846; Amonlirdviman, 2005 #870}, a modeling methodology originated from control engineering, is a good candidate. It allows both discrete and continuous models to interact and simulate simultaneously. Therefore, detailed continuous models such as ordinary

differential equations {Turing, 1990 #836} and stochastic master equations {Arkin, 1998 #793; Gillespie, 2007 #860; McAdams, 1997 #843}, and abstract discrete models such as qualitative modeling {Kuipers, 1986 #847} and phenomenological models can be combined to construct a holistic system model that is able to simulate the dynamic behavior of an entire biological system {Bower, 2001 #848; de Jong, 2002 #849; Voit, 2000 #850}. Chapter 3 describes our approach to model the *Caulobacter* cell cycle as a hybrid system and the validation of this model is given in Chapter 4.

1.2 Analysis of robustness

Mathematical models are great for simulation the dynamic behaviors of regulatory systems. However, models are mostly built upon experimental observations made under typical lab conditions, under which many contingency pathways are not activated. 73% of the 6000 genes in the budding yeast *Saccharomyces cerevisiae* have shown to be nonessential by gene depletion mutations {Dwight, 2004 #851; Giaever, 2002 #852}. These nonessential genes are likely to play an essential role in organisms' diverse natural habitats, especially when facing environmental or genetic perturbations {de Visser, 2003 #797; Alon, 1999 #789; Goulian, 2004 #801; Kitano, 2004 #807; Kitano, 2007 #808}, but their roles are often underappreciated or even neglected in models built on data collected in regular lab conditions. Without knowing the specific conditions triggering the expression of these nonessential genes, it is difficult to design the right experiments to identify their contributions to the entire system.

A well design engineering system has many considerations in addition to performing its essential tasks. For instance, to ensure robust operation of a silicon chip, certain design guidelines stipulates that additional circuitry should be included to react to hazardous conditions. We show in Chapter 5 that certain engineering tools developed to test and analyze the robustness of digital systems can be applied to models of biological systems as well. These tools helped us examine the robustness of such model {Savageau, 1971 #812} to discover potential mechanisms that contribute to the robustness of the overall system, thus enhancing its evolutionary competitiveness. By revealing how seemingly silent or redundant pathways are actually important, robustness analysis sheds light on the architecture, or “design” of regulatory systems

{von Dassow, 2000 #822; Von Dassow, 2002 #823; Ingolia, 2004 #805}. Such knowledge helps future endeavor in genetic engineering and synthetic biology.

1.3 *Caulobacter* cell cycle regulation

The organism modeled in this work is *Caulobacter crescentus*, a Gram-negative bacterium, and our model focuses on its cell cycle regulation. The cell cycle, or cell-division cycle, is the series of events that take place in a cell leading to its replication and division {Smith, 1973 #853}. Regulation of the cell cycle ensures orderly executions of these events and provides checks to prevent uncontrolled cell division.

Caulobacter is a model for studying bacteria cell cycle regulation, since its asymmetrical cell division allows the growth of synchronized cell population for identifying cell cycle regulated factors. The *Caulobacter* cell cycle division goes through three distinct phenotypic stages: swarmer, stalked, and pre-division (Figure 1.2) to generate two morphologically different daughter cells: a motile swarmer daughter cell and a sessile stalked daughter cell (Figure 1.2). During the cell cycle progression, the cell has to complete complex molecular functions like flagellum formation, DNA replication and cytokinesis, each of which takes significant amount of time and has to be processed in parallel to maximize growth rate. These processive cell functions make the cell cycle directional; that is, it is impossible to “reverse” the cell cycle once it starts.

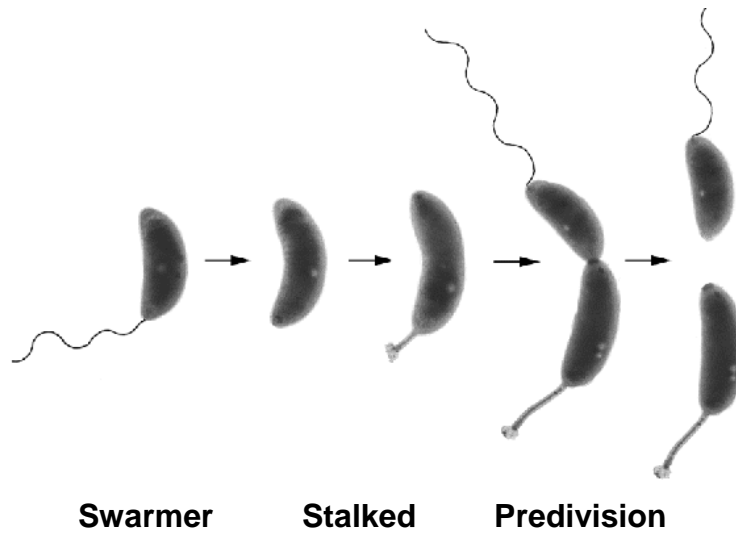


Figure 1.2 *Caulobacter* cell cycle progression

A network of protein regulators regulate these functions to make sure that the molecular events that control the cell cycle are ordered and do not go out of control {Laub, 2007 #854; McAdams, 2003 #58}. Multiple feedback control loops are formed by the transcriptional and non-transcriptional pathways in the regulatory network, which have been meticulously identified and characterized by various labs including the Lucy and McAdams lab at Stanford University. The *Caulobacter* cell cycle regulation is a complex process embodying many of the challenges discussed in the previous sections {McAdams, 2003 #58}. The next chapter reviews what is currently known about the *Caulobacter* cell cycle, on which the hybrid model described in chapter 3 is based.

Chapter 2

Caulobacter cell cycle regulation

Caulobacter crescentus is a Gram-negative, oligotrophic bacterium often found in nutrition-poor environments such as fresh water lakes and streams. Besides playing an important role in the carbon cycle, *Caulobacter* is a model organism for studying bacterial cell cycle regulation and asymmetrical division {McAdams, 2003 #58; Laub, 2007 #854}. *Caulobacter* cells divide asymmetrically, producing morphologically distinct daughters (Figure 1.2). One is a mobile "swarmer" cell that has a flagellum for swimming. The other is called a "stalked" cell because it has a long tubular stalk structure protruding from one pole that enables the cell to adhere to surfaces. A swarmer cell has to first shed its flagellum and grow a stalk to transform into a stalked cell before chromosome replication and cell division could begin. After division, the newly divided swarmer cell will swim away to search for new nutrient sources while the stalked daughter cell stays and keeps dividing. Therefore the dimorphic cell cycle of *Caulobacter* probably provides an advantage to compete in nutrient scarce environments.

2.1 cell cycle stages

The *Caulobacter* cell cycle progresses through three distinct stages: swarmer, stalked, and pre-division (Figure 2.1A), during which the cell sheds its flagellum, grows a stalk, replicates its DNA, and divides. The swarmer daughter cell has a single polar flagellum, polar chemotaxis receptors, and polar pili, and it cannot initiate DNA replication until after the period of motility is

completed when swarmer-to-stalked cell differentiation is initiated. Swarmer-to-stalked cell differentiation involves the loss of the flagellum and the polar chemotaxis receptors, retraction of the pili, construction of a stalk at the cell pole previously occupied by the flagellum, and initiation of chromosome replication. In contrast, the stalked daughter cell initiates chromosome replication immediately after cytoplasmic compartmentalization during cytokinesis. This cytoplasmic compartmentalization event occurs after decatenation of the replicated chromosomes when the constriction of the FtsZ-ring leads to fission of the inner membrane that separates the cytoplasm into two distinct chambers about 20 min before completion of cell division {Goley, 2007 #855; Collier, 2007 #856}. The two daughter cells have identical genotypes, but different morphology cell fates.

Cytoplasmic compartmentalization is the event that initiates the divergent genetic programs in each chamber, thus the next cell cycle effectively begins in each compartment at the instant of cytoplasmic compartmentalization, which is well before cell division {Goley, 2007 #855; Laub, 2007 #854; Chen, 2007 #857}

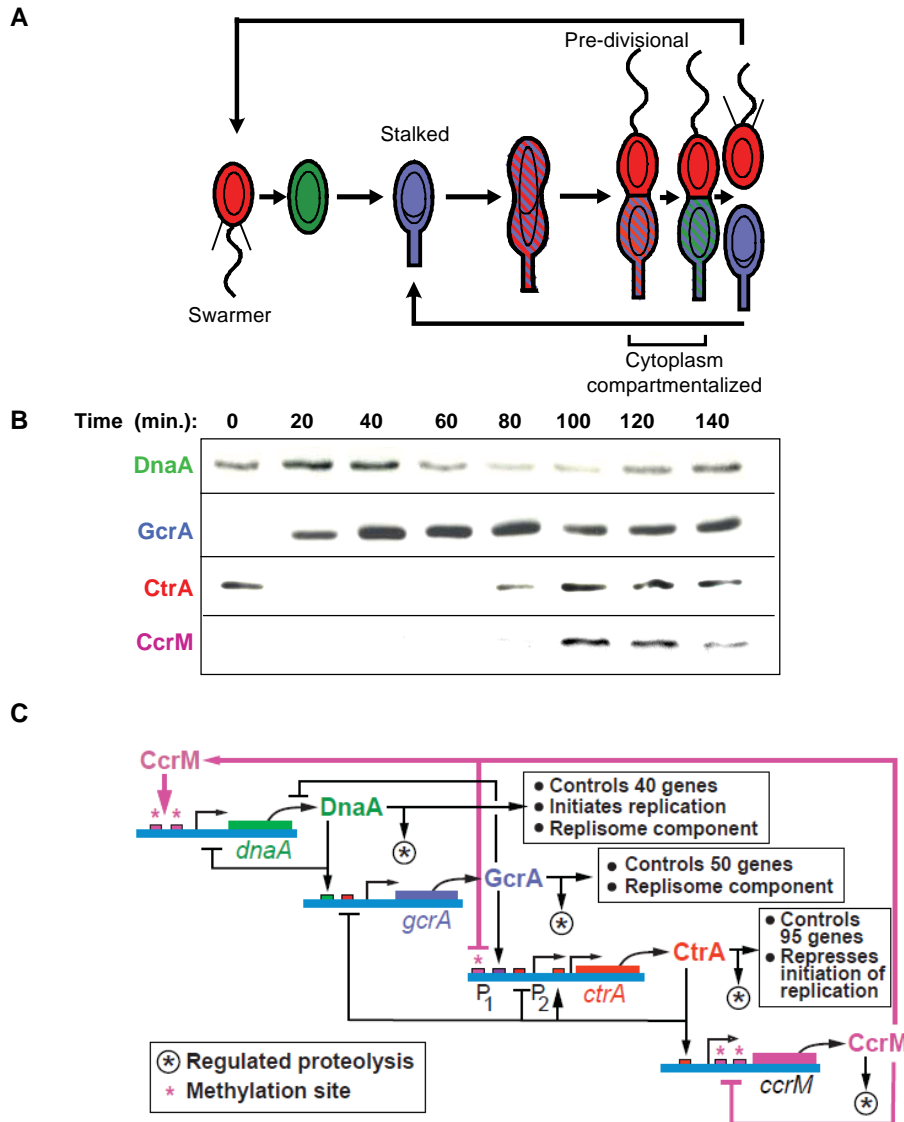


Figure 2.1 Genetic circuit that drives cell cycle progression. (A) Schematic of the *C. crescentus* cell cycle showing changes in master regulatory protein concentrations that control activation of numerous modular functions that implement the cell cycle. Predivisional cells are compartmentalized about 20 min before cell separation {Judd, 2005 #858} (B) Western blots showing concentrations of the master regulatory proteins during the cell cycle {Collier, 2007 #784; Collier, 2006 #783}. (C) Cascade of master regulators. DnaA, GcrA, CtrA, and CcrM form a cascade that control modules of cell cycle genes to drive the cell cycle forward.

2.2 Master regulatory network

Caulobacter has become an important model for studying the regulation of bacterial cell cycle because the lighter swarmer daughter cells can be separated from a mixed population in the

centrifuge to grow as a new synchronized cell population, which can then be sampled at different time points during the synchronized cell cycle progression to measure the temporal changes of various expression levels and distinguish cell cycle regulated proteins (Figure 2.1B). Groups or modules of proteins are synthesized during different stages of cell cycle progression to coordinate and perform modular functions such as chromosome replication and cytokinesis (Figure 2.1B). On top of the modules, a few master regulators form a central control circuit that switch these modular cell functions on or off in an orderly succession. These master regulators are transcriptional factors that regulate many downstream genes; that is, when the concentration levels of these regulator proteins in the cell become sufficiently high, it is statistically significant enough for the protein molecules to bind to the promoter regions of the downstream genes to either activate or repress their expression, which is the rate of mRNA transcription. mRNA serves as a template for protein synthesis, thus the levels of proteins translated from the downstream genes are controlled by these upstream master regulators.

Molecular-level characterization of regulatory pathways controlling the *Caulobacter crescentus* cell cycle is progressing rapidly so that a nearly complete system level description of the control system is now possible. The cyclical genetic circuit comprised of the CtrA, GcrA, DnaA, and CcrM master regulatory proteins directly controls the temporal expression of over 200 genes {Laub, 2002 #131; Holtzendorff, 2004 #33; Hottes, 2005 #782; Collier, 2007 #856} (Figure 2.1C). These proteins are present in succession as the cell cycle progresses (Figure 2.1A and Figure 2.1B). The cascade of regulatory factors starts with DnaA accumulation at the swarmer-to-stalked cell transition. DnaA is a replication initiation factor which promotes the unwinding of the double stranded DNA at *oriC* to initiate chromosome replication. DnaA also activates the transcription of multiple genes involved in DNA replication and cytokinesis as well as turning on the next gene in the cascade, *gcrA* (Figure 2.1C) {Hottes, 2005 #782; Collier, 2006 #783}. GcrA regulates genes involved in chromosome replication and segregation and turns off *dnaA* as it activates *ctrA* transcription {Holtzendorff, 2004 #33}. In addition to binding to the five DNA binding sites that overlap with the binding sites of the replication initiation protein and thus inhibiting initiation of DNA replication, CtrA directly controls the transcription of genes required for polar organelle biogenesis and cytokinesis, while turning off the transcription of *gcrA* and activating the synthesis of the CcrM DNA methyltransferase {Reisenauer, 1999 #233}.

This cascade of top-level master regulators creates the forward-biased, cyclical genetic circuit – the core cell cycle engine – that organizes cell cycle progression.

The core engine activates the different modules of proteins that implement the cell cycle, including several complex processive reactions that take extended time intervals to complete, particularly replication of the chromosome and cytokinesis as shown in the top half of Figure 2.2. The *Caulobacter* cell cycle control maximizes parallelism to shorten the required cell cycle time during growth phase. For example, *C. crescentus* DNA replication takes about 80 minutes in a typical lab condition. Even though compartmentalization of the inner membrane has to happen after the completion of chromosome replication and the decatenation of the newly replicated strands, the FtsZ ring, precursor to cytokinesis, starts forming at the division plane during chromosome replication and has to depend on a checkpoint to prevent premature compartmentalization {Degnen, 1972 #708}.

2.3 Feedback mechanisms

In *Caulobacter*'s natural habitat, due to environmental diversity and scarcity of nutrients, durations of these extended processive reactions vary significantly. To be in sync with the cell cycle, expression of the cascaded master regulators has to be tightly coupled to these processive reactions. Hence instead of being a stand-alone oscillator with a fixed period, the core engine has to be a control circuit with feedback signals to sense the processive reactions and express the master regulators “just in time”. Research has revealed three molecular feedback mechanisms for the regulatory network to sense the processive reactions. These mechanisms are methylation, phosphorylation, and proteolysis (Figure 2.2).

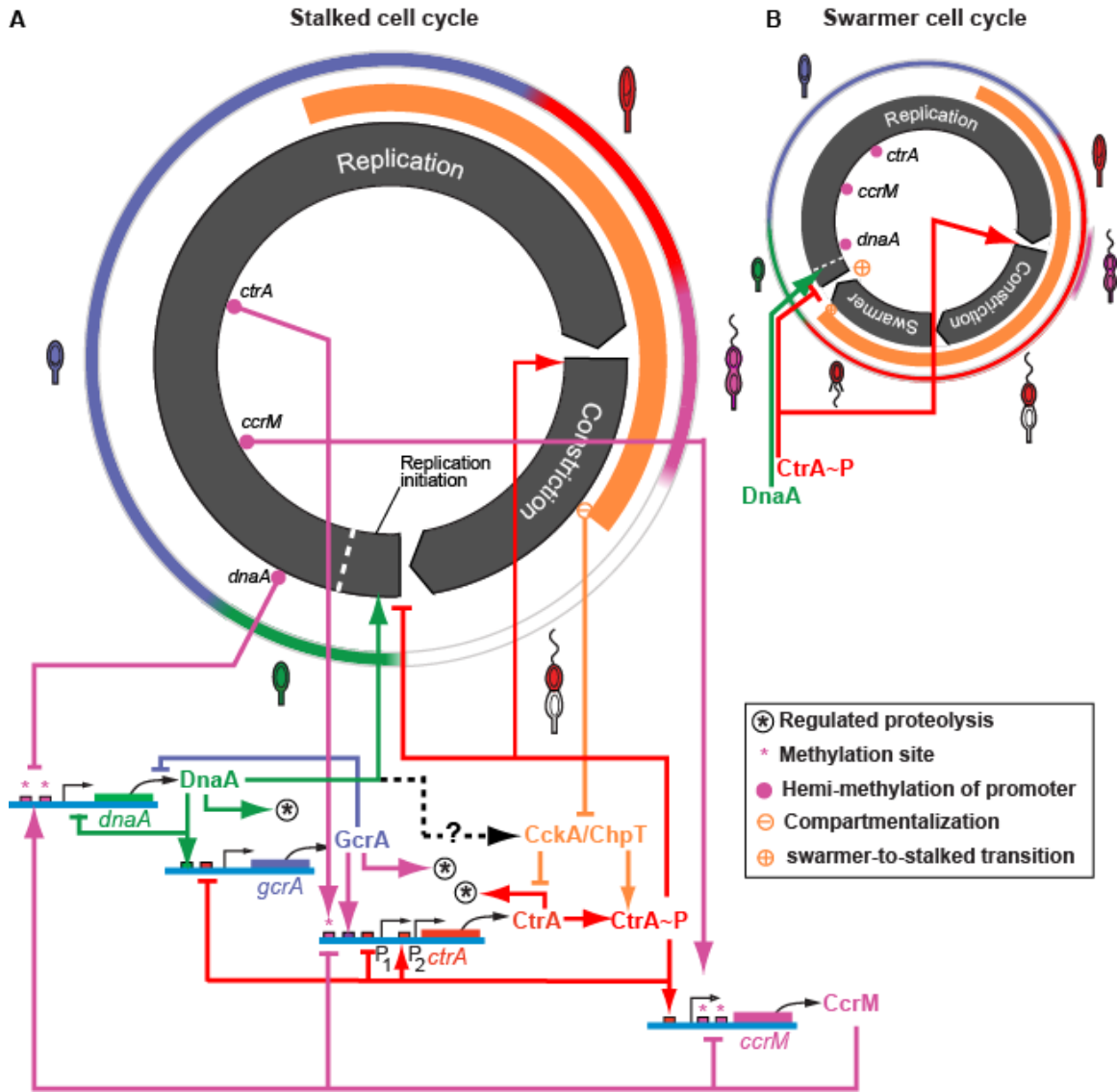


Figure 2.2 Schematic of the regulatory interconnections between the core cell cycle engine and cell cycle progression. (A) The duration of DNA replication and FtsZ-ring constriction in the cell cycle are approximately to scale. The core cell cycle engine below controls activation of the processive reactions that implement DNA replication and cell constriction, and feedback signals from these controlled processes synchronize the engine with their progression. Asterisks indicate CcrM methyltransferase target sites where the methylation state of the promoter region affects promoter activity {Collier, 2007 #784; Stephens, 1995 #350; Reisenauer, 2002 #104}. Cell stage is indicated by the cell-type icons on the perimeter. The outer circular band indicates by color coding the intervals of peak presence of DnaA, GcrA, CtrA, and CcrM. The orange arc on the next band indicates the interval when the CckA/ChpT pathway is active. Regulatory pathways are color coded by type: Red: CtrA~P. Blue: GcrA. Green: DnaA. Orange: CckA/ChpT phosphosignaling. Purple: DNA methylation-state regulation of promoter activity. (B) The swarmer cell cycle has an additional motile phase after cell division. Chromosome replication remains repressed in the swarmer cell

Methylation

The forward-biased cyclical cascade comprising the core DnaA/GcrA/CtrA/CcrM cycle is tightly coupled to chromosome replication through methylation (Figure 2.2). *C. crescentus* DNA replication involves two parallel reaction cascades, each with about two million reactions (the chromosome has about four million nucleotides), executed at two replication forks. The level of expression of three of the four cell cycle master regulator proteins, CcrM, CtrA and DnaA, is coupled to the progression of DNA replication by the DNA methylation-state change that occurs upon passage of a replication fork over their respective genes {Collier, 2007 #784; Stephens, 1995 #350; Reisenauer, 2002 #104}. The *dnaA* gene is transcribed preferentially from a fully methylated promoter. This methylation state control of *dnaA* transcription is of particular interest owing to DnaA's central role in the initiation of chromosome replication. The *dnaA* gene is near the chromosomal origin of replication (*Cori*), and upon passage of the replication fork it becomes hemimethylated, and thus down-regulated {Collier, 2007 #784}. The down-regulation of *dnaA* after the replication is initiated is significant to the robustness of the cell cycle because excessive amount of lingering DnaA might cause over-initiation of chromosome replication. The *ctrA* P1 promoter is activated when becoming hemi-methylated upon passage of the replication fork. Located approximately one-third of the way to *Cori*, the hemi-methylation of the *ctrA* P1 promoter indicates that chromosome replication is well under way so CtrA can be safely translated without the risk of blocking chromosome replication initiation. Furthermore, the newly synthesized CtrA molecules can prevent over-initiation of chromosome replication on the newly replicated strands as well as safely start the processes of flagellum formation on the swarmer pole and cytokinesis in parallel to chromosome replication, thus shortening the overall cell cycle duration. The enzyme that remethylates the entire DNA, CcrM, only accumulates near the completion of DNA replication, and it is then rapidly both deactivated and cleared from the cell {Wright, 1997 #285; Shier, 2001 #173}. Re-methylation of the chromosome by CcrM later enables *dnaA* transcription in preparation for the next cell cycle (In many bacteria, including *E. coli*, the DNA methylase is not cell cycle dependent.). CcrM also fully-methylates the *ctrA* P1 promoter to disable it after the positive auto-regulation loop of the *ctrA* P2 promoter takes over. The late timing of the expression of the *ccrM* gene is ensured by the preferential expression of the hemi-methylated *ccrM* promoter and its activation by the synthesis of CtrA. These methylation switches of gene expressions are utilized by *Caulobacter* to couple the expression of

the master regulators to different stages of chromosome replication, which assure that there is one and only one round of replication per cell cycle.

Phosphorylation and proteolysis

Asymmetric cell division depends on polar localized regulatory proteins and cytoplasmic compartmentalization, both of which are coupled to a sophisticated phospho-signal pathway involving CckA, DivK, PleC, ChpT, etc. The pathway rapidly switch on and off the master regulator CtrA through phosphorylation and rapid proteolysis {Domian, 1997 #290; Quon, 1996 #329; Biondi, 2006 #786} (Figure 2.3). This feedback mechanism causes CtrA to be abundant in the swarmer and predivision cell, but absent from the stalked cell where chromosome is being replicated.

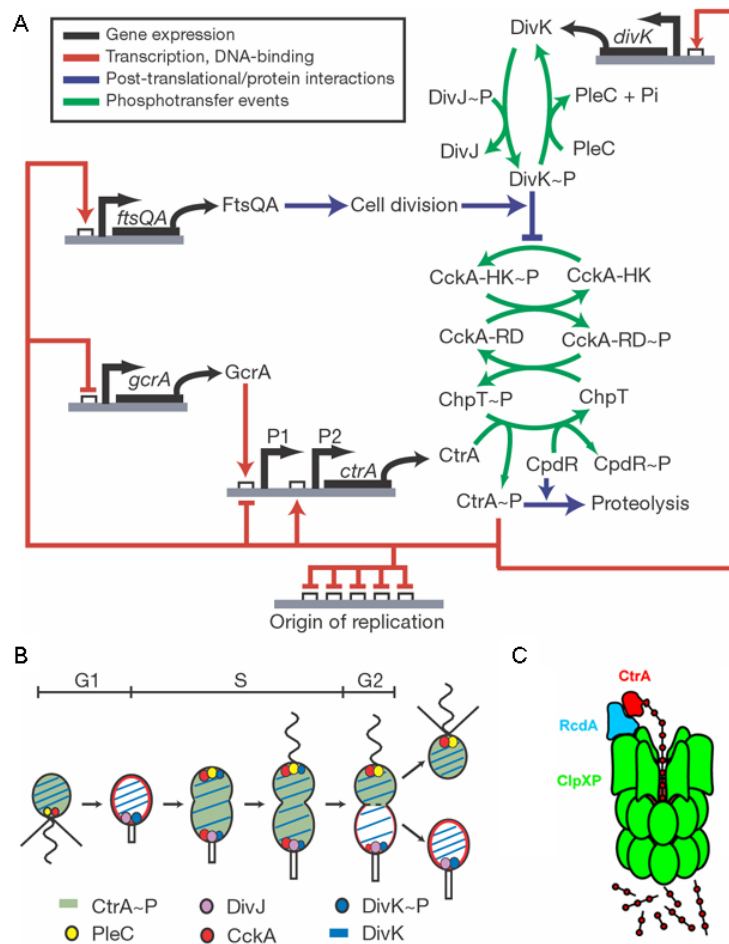


Figure 2.3 The phospho-signal pathway in *Caulobacter* cell cycle regulation. (A) Diagram of the integrated genetic circuit controlling cell cycle progression and cellular asymmetry in

Caulobacter crescentus. Biochemical relationships between components are colour-coded as indicated in the key {Biondi, 2006 #786}. (B) Summary of sub-cellular localization patterns for CckA, PleC, DivJ, CtrA~P, DivK and DivK~P during cell cycle progression {Biondi, 2006 #786}. (C) Proteolysis of CtrA molecules by ClpXP when RcdA is un-phosphorylated by CpdR through the phospho-pathway.

In the swarmer cell, during the 20 min period of motility after cell separation (Figure 2.2B), a sufficient amount of growth factor is accumulated and through a yet unknown pathway, causes the phospho-signal pathway to rapidly unphosphorylate and degrade CtrA simultaneously, thus starting the swarmer to stalked transition and the initiation of chromosome replication.

CtrA in the stalked daughter cell is rapidly depleted by the phospho-pathway right after compartmentalization to differentiate it from the swamer daughter cell which has high level of CtrA. The constriction of the FtsZ-ring that divides the cell is a complex cascade involving a changing set of many proteins {Margolin, 2005 #861}. Constriction is dependent on presence of the CtrA~P-activated FtsA and FtsQ proteins {Martin, 2004 #17; Wortinger, 2000 #200}. Constriction is first apparent late in chromosome replication and cell separation occurs about 25 min after completion of replication. In *C. crescentus* cells, there are two constrictive mechanisms late in cytokinesis, one for the inner, and one for the outer, cell membrane. The inner membrane fissions about 20 min before completion of outer membrane constriction to divide the cytoplasm into two compartments {Judd, 2005 #858; Judd, 2003 #69}. This cytoplasm compartmentalization event triggers elimination of CtrA~P in the nascent stalked cell compartment, which both enables activation of DNA replication and precipitates major changes to the transcriptome since CtrA~P directly regulates transcription of about 95 genes {Laub, 2002 #131; McGrath, 2006 #859; Iniesta, 2006 #785; Biondi, 2006 #786}. The dynamic localization of regulatory proteins and proteolytic subsystems to the cell poles is essential to asymmetric cell division {McAdams, 2003 #58; Collier, 2007 #856}. Immediately upon compartmentalization, differentiation begins owing to isolation of key phosphorylation dependent regulatory proteins from their cognate kinases {Iniesta, 2006 #785; McGrath, 2004 #35; Matroule, 2004 #19} and/or perhaps to differential sequestering of a phosphatase {Biondi, 2006 #786}. Large differences in binding affinity between the phosphorylated and unphosphorylated response regulators in the nascent daughter cell compartments cause gene expression profiles to diverge, and thus, differential development programs can proceed thenceforth, with profound consequences for the

fates of the two daughter cells. Cytoplasmic compartmentalization disrupts the distributed phosphosignaling network involving polar localized CckA histidine kinase and cytoplasmic ChpT phosphotransferase to trigger rapid elimination of activated CtrA~P from the nascent stalked daughter cell so that chromosome replication can initiate (Figure 2.3A,B) {Iniesta, 2006 #785; Biondi, 2006 #786; Jacobs, 1999 #248}. In contrast, elimination of CtrA~P from the swarmer cell is delayed until about 20 min after daughter cell separation when the CckA/ChpT pathway is disrupted by another mechanism. The distinctive identity of the subsequent daughter cells, each containing one of the chromosomes of the predivisional cell, begins at the instant of cytoplasmic compartmentalization {Judd, 2003 #69}.

2.4 system model

In this chapter, we outlined the molecular mechanisms that comprise the *Caulobacter crescentus* cell cycle control system. The details of the control circuitry have been characterized by many laboratories over several decades. Table 2.1 provides a roadmap to these papers and indicates where the molecular and genetic mechanisms of the key proteins in the model are characterized relating to the key proteins in the model.

Table 2.1 Experiments done with *Caulobacter* cells and conclusions used to construct the model

Protein	Refs.	Conclusion used to construct the model
DnaA	{Gorbatyuk, 2001 #168}	DnaA is necessary for the initiation of DNA replication
DnaA	{Hottes, 2005 #782; Collier, 2006 #783}	DnaA activates <i>gcrA</i> transcription
DnaA	{Hottes, 2005 #782}	DnaA activates <i>ftsZ</i> transcription
DnaA	{Hottes, 2005 #782}	DnaA activates <i>dnaB</i> transcription
DnaA	{Gorbatyuk, 2005 #754}	DnaA is subject to cell cycle-regulated proteolysis
DnaA	{Collier, 2007 #784}	<i>dnaA</i> transcription is activated when the <i>dnaA</i> promoter is fully-methylated
FtsZ	{Wang, 2001 #169}	FtsZ is necessary for cell constriction
FtsZ	{Kelly, 1998 #267}	FtsZ is subject to cell cycle-regulated proteolysis
FtsA	{Osley, 1977 #658}	FtsA is necessary for cell constriction

FtsA	{Martin, 2004 #17}	FtsA is probably subject to cell cycle-regulated proteolysis
FtsQ	{Martin, 2004 #17}	FtsQ is subject to cell cycle-regulated proteolysis
CcrM	{Zweiger, 1994 #373}	CcrM is necessary for the methylation of the chromosome
CcrM	{Wright, 1996 #318}	CcrM is subject to cell cycle-regulated proteolysis
CcrM	{Stephens, 1995 #350}	<i>ccrM</i> transcription is probably repressed when the <i>ccrM</i> promoter is fully-methylated
CtrA	{Quon, 1998 #274}	CtrA is necessary to block the initiation of DNA replication
CtrA	{Kelly, 1998 #267}	CtrA represses <i>ftsZ</i> transcription
CtrA	{Wortinger, 2000 #200}	CtrA activates <i>ftsQA</i> transcription
CtrA	{Holtzendorff, 2004 #33; Collier, 2006 #783}	CtrA represses <i>gcrA</i> transcription
CtrA	{Domian, 1999 #239}	CtrA represses <i>ctrA</i> transcription from the <i>ctrAP1</i> promoter
CtrA	{Domian, 1999 #239}	CtrA activates <i>ctrA</i> transcription from the <i>ctrAP2</i> promoter
CtrA	{Reisenauer, 1999 #249}	CtrA activates <i>ccrM</i> transcription
CtrA	{Domian, 1997 #290}	CtrA is subject to cell cycle-regulated proteolysis
CtrA	{Reisenauer, 2002 #104}	<i>ctrA</i> transcription is repressed when the <i>ctrAP1</i> promoter is fully-methylated
CtrA	{Domian, 1997 #290; Quon, 1996 #329}	CtrA needs to be phosphorylated to be active
GcrA	{Holtzendorff, 2004 #33}	GcrA activates <i>dnaB</i> transcription
GcrA	{Holtzendorff, 2004 #33}	GcrA activates <i>ctrA</i> transcription from <i>ctrAP1</i>
GcrA	{Holtzendorff, 2004 #33}	GcrA represses <i>dnaA</i> expression
GcrA	{Collier, 2006 #783}	GcrA is subject to cell cycle-regulated proteolysis
CckA	{Jacobs, 1999 #248}	The CckA signal activates the phosphorylation of CtrA
CckA	{Iniesta, 2006 #785}	The CckA signal represses the proteolysis of CtrA
CtrA	{Wortinger, 2000 #200}	CtrA is not synthesized in predivisional cells if replication is inhibited

Several qualitative descriptions of various aspects of *C. crescentus* control system circuitry were available {Laub, 2007 #854; McAdams, 2003 #58; Biondi, 2006 #786} and a stalked cell cycle subcircuit model was reported {Li, 2008 #862}. Partially due to the difficulty of handling

lack of complete information and levels of abstraction, the subcircuit model did not include phosphorylation of CtrA by the phospho-signal pathway, which is critical for the feedback aspect of the control circuit.

We wanted to develop a scalable simulation of control of the coupled swarmer and stalked cell cycles with emphasis on handling different levels of abstraction. The simulation model needed to predict the progress of the regulatory machinery into either compartment of the predivisional cell to shed light on the events related to asymmetry, because it was difficult to separate the two daughter compartments *in vivo* to observe them separately. It was necessary for the complete system model to include the key regulatory proteins as well as the lengthy molecular processes such as chromosome replication and cytokinesis to capture the dynamic behavior of the entire feedback control. The next chapter describes our implementation of the cell cycle model as a hybrid system.

Chapter 3

Hybrid system modeling

A complete system model has to include both subsystems of the *Caulobacter* cell cycle regulation, the regulatory genetic circuit and the cell function processes, to explore the dynamics of the cell cycle feedback control (Figure 3.1). The protein components of the genetic circuit subsystem include the four master regulator proteins (DnaA, GcrA, CtrA, and CcrM) that comprise the core cyclical circuit and DnaB, FtsZ, and FtsQA (Figure 3.1B). DnaB, FtsZ, and FtsQA are components of pathways that connect the core engine with DNA replication and cytokinesis. This is a parsimonious model of the cell cycle control circuitry. For example, DnaB is only one of the proteins in the replication complex whose synthesis is activated by DnaA {Hottes, 2005 #782}. FtsQA represents two proteins, FtsQ and FtsA (whose genes are in an operon), required for initiation of cell constriction. The cell process subsystem includes phenomenological models of the progress of chromosome replication and cell constriction (Figure 3.2B). These two subsystems determine the timing of the changes in the methylation state of the *dnaA*, *ctrA*, and *ccrM* promoter regions and of cell compartmentalization {Collier, 2007 #784; Reisenauer, 2002 #104}.

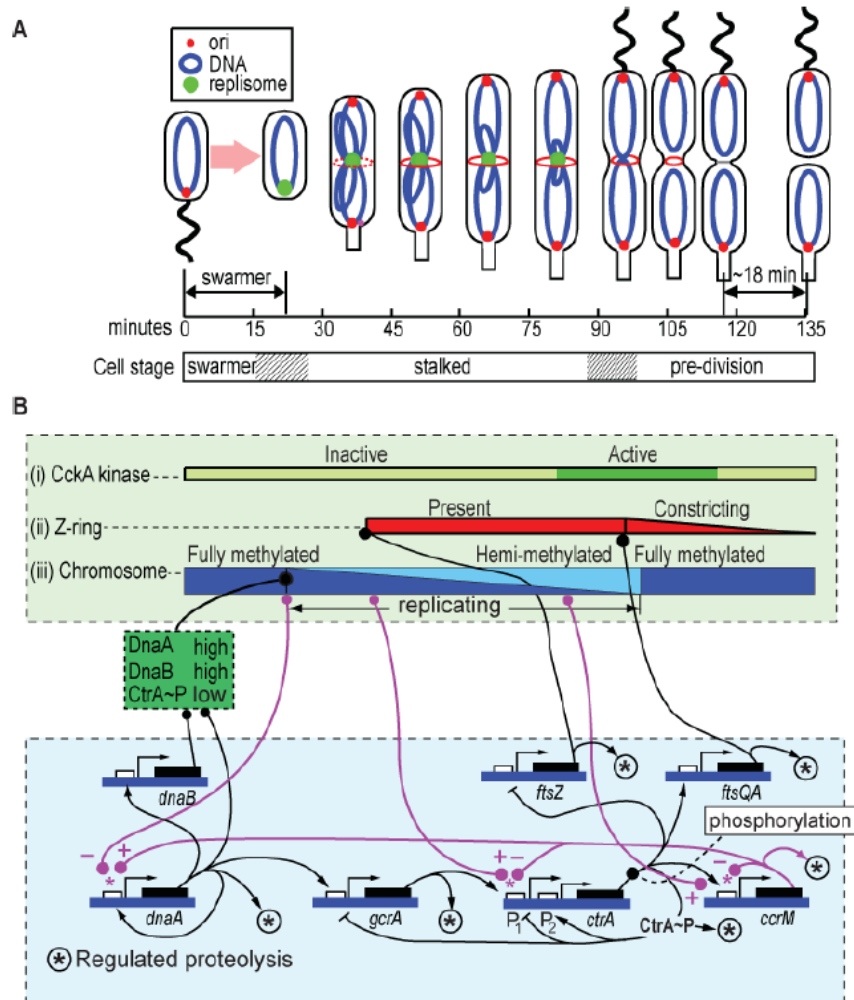


Figure 3.1 *C. crescentus* cell cycle control system. (A) The red FtsZ ring at mid-cell constricts to divide the cell. Inner membrane fission compartmentalizes the cytoplasm about 18 min before daughter cell separation {Judd, 2005 #858}. (B) Schematic of cell cycle control. Top green rectangle: (i) Observed timing of the CckA-originated phosphosignal activity. (ii) Timing of Z-ring appearance and constriction. (iii) Chromosome replication timing and methylation status. The bottom blue rectangle encloses the genetic circuit that drives cell cycle timing. The purple lines indicate effects of the DNA methylation state of sites in gene promoter regions (described in the text). For example, *dnaA* is maximally expressed when its promoter is fully methylated {Collier, 2007 #784}. Middle, dark green dashed rectangle: Conditions necessary within the cell cycle control system simulation to activate formation of the replisome and subsequent initiation of DNA replication. The *dnaB* gene is one of several replisome proteins components whose synthesis is activated by DnaA. Accumulation of FtsZ initiates formation of the FtsZ-ring. Expression of the *ftsQA* operon coincides with the initiation of the FtsZ-ring constriction.

To handle the complexity of cell cycle regulation, the system simulation model should include several features: (i) a top-down, hierarchical approach able to handle different levels of

abstraction; (ii) a realistic architecture that mimics the *in vivo* feedback design of the cell's control system (Figure 3.3A); (iii) a scalable organization designed for future incorporation of additional pathways; (iv) a flexible and parameterized design for *in silico* mutant simulation and in-depth analysis.

3.1 Ordinary differential equation (ODE)

With molecular level characterizations, it is now feasible to build a quantitative model of the *Caulobacter* cell cycle beyond previous qualitative descriptions. Ordinary differential equations (ODE) have been widely used for simulating the dynamics of regulatory systems by tracing protein and mRNA levels {Chen, 2004 #864; Novak, 2003 #865}, which enabled advanced analysis techniques like phase plane and bifurcation to explain observed behaviors {Novak, 2003 #865; Kim, 2006 #863}. For *Caulobacter*, the entire cell cycle genetic circuit governing regulation of protein synthesis, proteolysis and activation of the *Caulobacter* regulatory network can be modeled by ODEs. The level of promoter activation (as a fraction of the maximum activation) is modeled using functions based on a Hill function approach {Rosenfeld, 2005 #866; Rosenfeld, 2002 #867}. Protein production (nM/second) is modeled by a multiplicative constant representing the maximum synthesis rate, times the fractional promoter activation. This is equivalent to assuming a constant average rate of protein production per mRNA. The following equations are standard Hill functions for modeling the protein synthesis rate of a gene whose promoter is either activated (Eq. 3.1) or repressed (Eq. 3.2).

$$\frac{d[C]}{dt} = \frac{\beta/V}{1 + \left(\frac{[C_d]}{[C_t]}\right)^n} \quad (3.1)$$

$$\frac{d[C]}{dt} = \frac{\beta/V}{1 + \left(\frac{[C_t]}{[C_d]}\right)^n} \quad (3.2)$$

where n is the Hill coefficient reflecting the cooperativity of the transcriptional factor, $[C_d]$ is the concentration of the transcriptional factor that yields half-maximal expression, $[C_t]$ is the concentration of the transcriptional factor, β is the maximal protein production rate, V is the volume of a *C. crescentus* cell, and $[C]$ is the concentration of the protein molecules synthesized from the regulated gene.

Even though transcription and translation are innate stochastic processes {Rosenfeld, 2005 #811}, in this approximation, stochastic effects are neglected in the model which assumes that the averaging effect of the number of protein synthesized in the time scale relevant to cell cycle makes the stochastic effects insignificant. Because with a short half-life transcribed mRNA levels usually reach equilibrium quickly and the rate of translation is proportional to the mRNA level, protein synthesis rate is proportional to the mRNA level in our models.

Natural degradation or proteolysis of a protein is modeled by an exponential decay function with a half-life parameter:

$$\frac{d[C]}{dt} = -\lambda \cdot [C] = -\frac{\ln 2}{hl} [C] \quad (3.3)$$

where hl is the half-life of the protein in min, and λ is the degradation rate constant in min^{-1} .

When combined, the above ODE equations describe the dynamics of various protein regulators in the regulatory system (Figure 3.2). The simulated protein and mRNA (proportional to the protein synthesis rate) levels can be compared to Western blot and Microarray measurements for validation purposes. Protein degradation rate, which is indicated by the half-life parameter, can be measured by pulse-labeled immunoprecipitation.

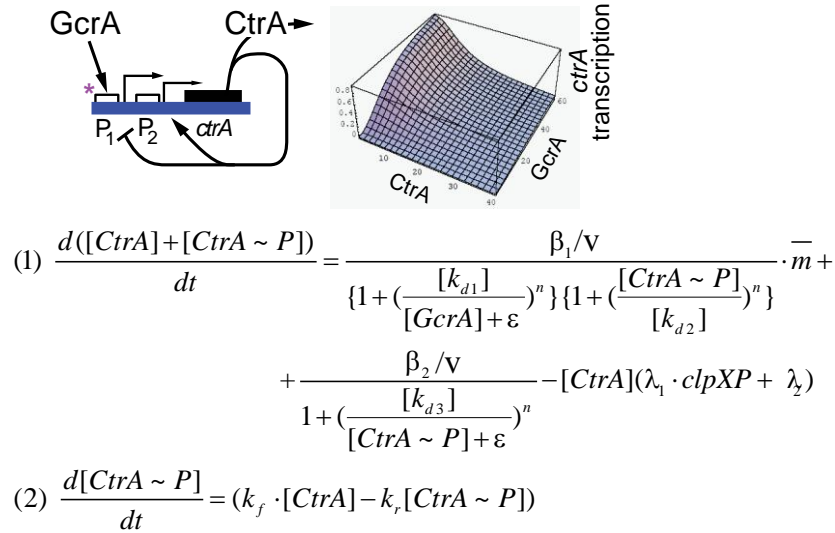


Figure 3.2 A simplistic model of the dynamics of the CtrA level. The *ctrA* P1 promoter is activated by GcrA, repressed by CtrA~P, the phosphorylated form of CtrA and is repressed when fully-methylated ($m=1$). The *ctrA* P2 promoter is activated by CtrA~P, forming a positive feedback loop. CtrA degrades slowly, but can be rapidly degraded by the proteolytic ClpXP complex ($clpXP=1$). The rate constants of phosphorylation of CtrA are decided by the presence of kinases.

3.2 Hybrid system

Despite all the successes ODE models have achieved, they have limitations when dealing with biological systems. To continuously track their dynamic behaviors, ODE models require detailed characterizations of the modeled biological processes with a great number of parameters, but most biological systems include processes that are not well characterized. In the case of the CckA/ChpT phosphosignaling pathway (Figure 2.3), mechanisms of the pathway are not completely identified, but the function of the pathway and the timing of its operation within the cell cycle are well characterized {Iniesta, 2006 #785}. Similarly, the time from onset to completion of DNA replication and the time from initial cell constriction to compartmentalization are known {Keiler, 2003 #91}. In these cases, building detailed ODE models is infeasible, nevertheless, including phenomenological, but functionally accurate, models with correct cell cycle timing is sufficient to holistically simulate *Caulobacter* cell cycle regulation.

In the *Caulobacter* cell, there also exist fast reactions that act more like switches, which happen almost instantaneously compared to protein synthesis. For instance, Cytoplasmic compartmentalization is essentially instantaneous {Judd, 2005 #858}. The promoter methylation state changes when the replication fork traverses promoters that involve this mechanism {Collier, 2007 #784} are also instantaneous, as is the remethylation reaction. Each of these events triggers discrete changes in the network of gene transcription, phospho-signaling, and protein-level reactions that control the cell cycle. While the exact mechanistic models of these reactions are not available, “making up” continuous ODE models for these switch-like events tends to over-complicate the system model; rather, reactions that occur essentially instantaneously should be modeled as discrete switching events, which leads to a hybrid system model.

Concepts of hybrid system in control engineering help extend dynamic system models beyond the limitations of continuous ODE models. A hybrid system is a dynamic system that exhibits both continuous and discrete dynamic behavior — a system that can both *flow* (described by a differential equation) and *jump* (described by a difference equation) {Branicky, 1998 #868; Alur, 2002 #790}. A typical instance in engineering is a real-time system where physical processes such as thermal and chemical reactions are controlled by embedded digital controllers. A hybrid system has the benefit of encompassing a larger class of systems within its structure, allowing for more flexibility in modeling dynamic phenomena{Lincoln, 2004 #883}. Much effort has been put into developing efficient software tools for modeling and performing formal verification for safety and stability analysis on hybrid systems{Amonlirdviman, 2005 #870; Alur, 1993 #869}.

In general, a hybrid system can be described by a few pieces of information. The *state* of the system consists of vector signals, which can change according to dynamic laws in the system *data*. The data includes a *flow equation*, $f(x)$, which describes the continuous dynamics, a *flow set*, C , in which flow is permitted, a *jump equation*, $g(x)$, which describes the discrete dynamics, and a *jump set*, D , in which discrete state evolution is permitted.

The organization of the *Caulobacter* cell cycle feedback control *in vivo* can be partitioned into two subsystems: the cell cycle regulatory network and the lengthy cell processes (Figure 3.3A). The regulatory network expresses different protein levels to control the cell processes; in

return when the processes reach certain distinct phases, feedback mechanisms such as methylation and the phospho-signal pathway act as fast switches to turn on or off gene expressions swiftly. These two subsystems can naturally be modeled as a hybrid system: continuous ODE equations, being the flow equation $f(x)$, describe the well characterized regulatory network subsystem; successions of discrete states (called a state machine in engineering dialect), being the jump equation $g(s)$, describe the subsystem of fast reactions and phenomenological processes with distinct phases (Figure 3.3B).

Modeling *Caulobacter* cell cycle regulation as a hybrid system provides the system model with the capability of dealing with different levels of abstraction{Lincoln, 2004 #883}. Not well characterized or even hypothetical reactions can be included in the model as discrete phenomenological processes until further details are known. By integrating both subsystems into a single holistic simulation model, we are able to explore the dynamics of this close-loop feedback control.

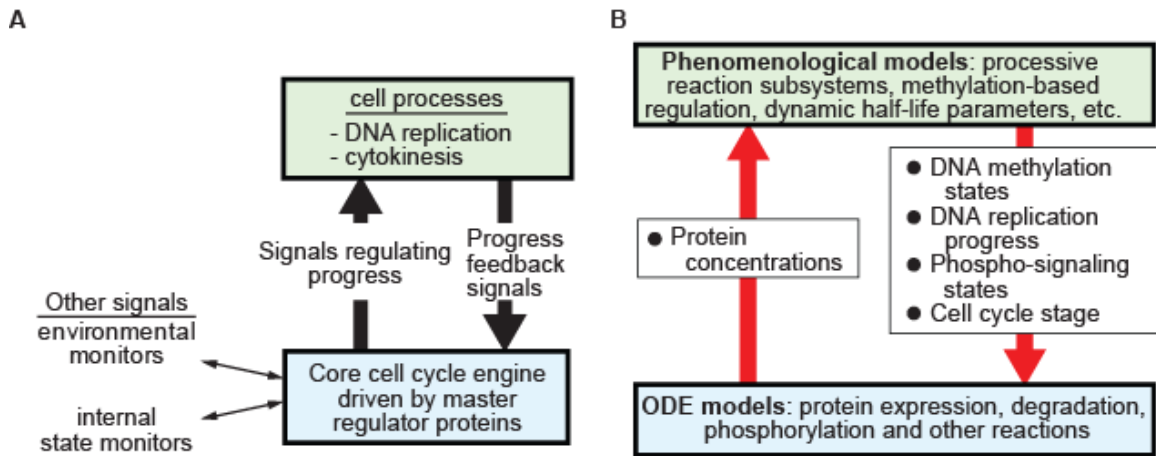


Figure 3.3 Simulation model overview. (A) Organization of cell cycle control. The core cell cycle engine activates modular functions (e.g., chromosome replication and cytokinesis). Feedback signals from these functions pace progression of the engine. Internal and external signals feed into the engine and can also slow or stop the cell cycle. (B) The simulation model architecture mirrors organization of the cell cycle control system. The ODE module models the parts of the system that are described by ordinary differential equations (e.g., kinetic reaction equations for protein synthesis and degradation). The phenomenological models include progression of processive modular reaction systems (DNA replication and cytokinesis) and they generate feedback signals (e.g., methylation state of the *ctrA* and *dnaA* promoter regions) that change or update parameters in the Simulink integration. Phenomenological models also generate signals tied to simulation time or to conditional events within the simulation. Examples

include the ON or OFF state of the CckA-originated phosphosignal, satisfaction of conditions for activation of DNA replication, and the current cell stage (swarmer, stalk, or predivisional).

3.3 Model implementation in Matlab

The simulation model has a hierarchical architecture that mimics the organization of the cells regulatory control system (Figure 3.3). The simulation is constructed using the Matlab Simulink {Mathworks #871} and Stateflow {Mathworks #872} tools that are widely used by control engineers to design, analyze, and simulate control systems. Simulink is a Matlab-integrated platform for simulation and design of dynamic systems with an interactive graphical environment. Stateflow is tightly integrated with Matlab and Simulink, and it is used to model discrete event-triggered changes in simulation parameters during progression of the simulation. The combination of Simulink and Stateflow gives an interactive simulation tool well suited for modeling of hybrid dynamic systems, that is, systems that include some elements describable by ordinary differential equations (ODEs) and other elements that are based on discrete states. This combination of features is well matched to requirements for simulation of the *C. crescentus* cell cycle control system. In addition, the modular architecture of models constructed with Simulink and Stateflow will facilitate extension of the *C. crescentus* cell cycle model to add additional mechanistic details as they are reported and to extend the cell cycle model to incorporate environmental sensor/response systems that affect operation of the cell cycle. Thus, this extensible modeling paradigm provides an approach applicable to construction of a whole cell model. The simulation files are available at:

<http://www.stanford.edu/group/caulobacter/CellModel>.

The top level Matlab model file (Caulobacter.mdl) includes two major blocks. The Simulink subsystem models the core oscillatory circuit comprised of DnaA, GcrA, CtrA, and CcrM regulatory proteins and a parsimonious additional set of proteins (DnaB, FtsZ, and FtsQA) that are in the pathway for control of two key controlled subsystems, DNA replication and cell constriction (Figure 3.4). The Stateflow subsystem monitors progress of the Simulink cell cycle simulation to detect conditional events, and it contains phenomenological models of the operation of the two controlled processive subsystems. For example, the conditions to initiate DNA replication in the model are (DnaA AND DnaB) NOT CtrA (Figure 3.1 and Figure 3.4). (This logic is an abstraction representing necessary, but not necessarily sufficient, conditions for initiation of replication.) The Stateflow subsystem monitors the changing levels of the various proteins modeled by the Simulink subsystem and detects satisfaction of the “initiate replication” conditions. When the “initiate replication” event is detected, the Stateflow model of DNA replication progression is initiated. Key outputs from this model are the timing (in the simulation) of replication of *dnaA*, *ctrA*, and *ccrM* genes. At the time of their replication, the promoter regions of these genes become hemimethylated. Their hemimethylation status is signaled to the Simulink model where it affects the rate of expression of these genes {Collier, 2007 #784}.

The Simulink system is an ordinary differential equation (ODE) solver with a graphical interface. Rates of protein synthesis and proteolysis and phosphorylation reactions within the regulatory protein circuit of the cell cycle engine are all modeled in Simulink as a system of ODEs. Principal outputs of the Simulink subsystem are estimated protein and mRNA levels versus time in the cell cycle. These changing protein levels and the cell cycle time are inputs into the Stateflow subsystem. The outputs of the Stateflow subsystem are values for binary switched parameters in the simulation model in the Simulink subsystem. Stateflow provides the capability to change parameter values in the ODEs in Simulink as the simulation progresses, that is, as the equations are being numerically integrated. As described above for the promoter methylation states and conditions for initiation of DNA replication, the Stateflow subsystem monitors the current (in simulation time) values of protein levels and switches the parameters when some condition is satisfied. Stateflow can also switch parameters at designated time points (in

simulation time) during simulation. Examples of switched parameter values provided to Simulink by the Stateflow subsystem include the DNA methylation state of methylation-dependent promoters and the cell stage (swarmer, stalked, or predivisional) that is used in Simulink to sets protein half-lives to the values experimentally observed in synchronized cell populations.

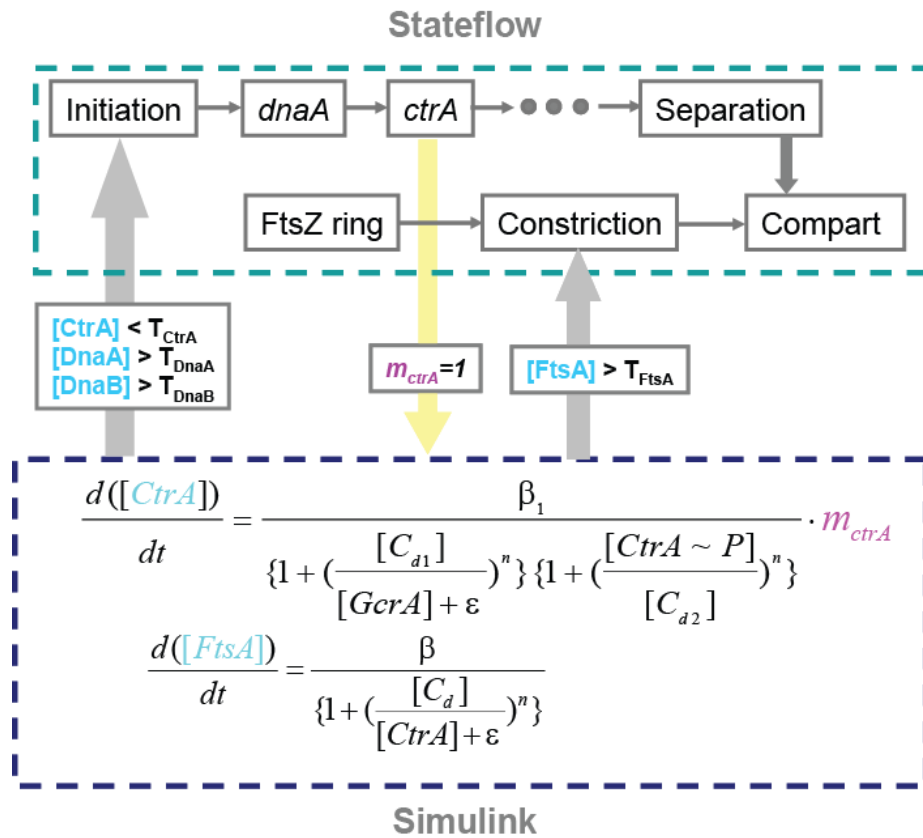


Figure 3.4 An illustration of the operation of the hybrid system model implemented in Matlab. The regulatory circuit is modeled using ODEs in the Simulink block on the bottom. The cell function processes (chromosome replication and cytokinesis) are modeled as discrete state machines in the Stateflow block on the top. The threshold levels for the protein regulators to initiate a cell process are indicated by the two boxes on top of the upward arrow. The feedback mechanism of methylation control is modeled as a binary switch indicated by the box on top of the downward arrow.

General modeling considerations

We are modeling events and chemical reactions as they occur in a single cell reaction chamber. As was mentioned earlier, the phenomena leading to stochastic reaction rates and variations in progress of cell cycle events are not considered in this cell level simulation. However, the effect of stochastic variations in rates of progression of chemical cascades on the *C. crescentus* cell cycle is to cause dispersion in timing of events over the cell population, which we can reflect by convolving the single cell predictions of the protein and mRNA concentrations profiles with a Gaussian function. The results of the convolution approximate the corresponding experimentally observed dispersion patterns from cell population measurements. These convolved protein and mRNA profiles can then be compared to experimental results from Western blots or micro-array mRNA assays on cell populations.

Models in the Simulink subsystem

Where we know genetic mechanisms and bio-chemical reactions, we use explicit or approximate kinetic models in the form of ODEs in the Simulink subsystem. Where we do not know details, but we do know phenomenology such as signal timing or the average time to complete a process, phenomenological models were constructed within the Stateflow subsystem. An example is the model of the CckA-originated phosphosignal that controls CtrA proteolysis and its phosphorylation state. From experimental observations, we know that this signal is interrupted at the time of cytoplasmic compartmentalization {Judd, 2003 #69; Domian, 1997 #290} and at the swarmer-to-stalked cell transition. We also know the timing of these two events and the timing of reactivation of the phosphosignal path in the predivisional cell from experimental observation. We define a conditional event in Stateflow to switch this CckA-originated phosphosignal off and on in the Simulink simulation at the appropriate times. (The robustness analysis described in Chapter 5 investigated sensitivity to specific timing of the CckA phosphosignal and found that the architecture of the cell cycle control circuit provides for successful completion of the cell cycle even when the specific timing of this signal varies over a wide range.) This phenomenological modeling approach enables realistic simulation of the known circuitry and phenomenology even though biochemical mechanisms in the pathway are still incompletely characterized.

Table A.1 in Appendix A shows the ODE models of all the protein regulators and the activation of their respective promoters. The effects of promoter methylation states are included in the promoter activation models, as are the cases where there are multiple promoters or multiple regulatory ligands. Instantaneous values for the binary switch parameters in the ODEs (e.g., the methylation state of a promoter) are determined by the Stateflow subsystem. The timing of initiation of CtrA proteolysis is determined by the Stateflow controlled phosphosignaling pathway originating at CckA. In other cases where there is experimental data for different half-lives at different cell cycle stages, the respective half-life parameters are set to the observed values (by input from the Stateflow subsystem) as the cell cycle progresses.

The model predicts the changing intracellular concentration of the regulatory proteins and mRNAs. In the model equations (Table A.1, Appendix A), protein and mRNA rates are in nM/sec. Where necessary conversions between molecules/sec per cell and nM/sec per cell were made using

$$M \text{ molecules/sec} = \frac{M}{A_g \cdot V_{avg}} \text{ nM/sec} \quad (3.4)$$

where A_g is Avagadro's number in nmol^{-1} and V_{avg} is the average value over the cell cycle of the *Caulobacter* cell volume in liters. Using 7.5×10^{-16} liters ($0.75 \mu\text{m}^3$) for V_{avg} , the conversion factor $\frac{1}{A_g \cdot V_{avg}}$ is 2.2. Since we simulate the molecular concentrations instead of the number of molecules/cell at the time of compartmentalization, concentrations of cytoplasmic proteins are initially equivalent in each of the new compartments. However, polar localized proteins can have significantly different concentrations in each compartment, which can differentially affect the subsequent evolution of the respective biochemical and genetic systems of the compartments.

We compare our simulation prediction to experimental values obtained from Western blots. Each time point of the Western blots was normalized to the same cell mass (OD660nm) to facilitate comparisons with the *in silico* simulation predictions.

Models in the Stateflow subsystem

The Stateflow subsystem includes models of four physical processes: cell stage, the CckA phosphosignal state (i.e., ON or OFF), the progression of chromosome replication, and the progression of cytokinesis (Figure 3.5). Chromosome replication and cytokinesis are processive models, that is, complex biochemical reactions that take extended time intervals to complete. The chromosome replication model computes the fractional completion of replication as a function of time after initiation, assuming a linear rate of replication. This model also determines the DNA methylation state of the promoters whose activity is affected by methylation and provides a corresponding input signal to the genetic circuit model and the Simulink subsystem. The fractional completion of cytokinesis is also modeled with a linear model that signals when cytoplasmic compartmentalization occurs and when daughter cell separation occurs.

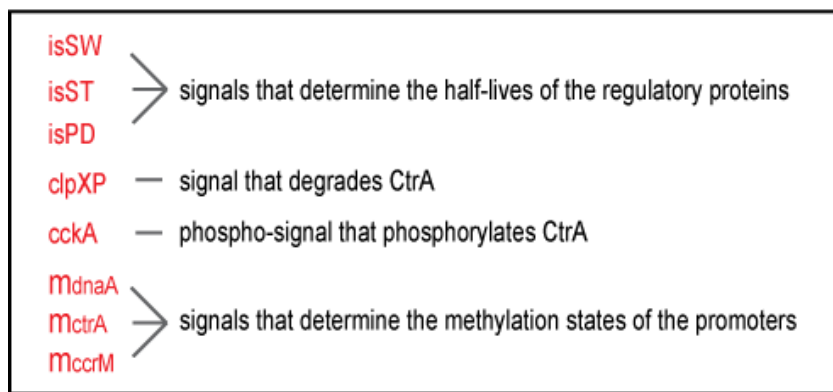
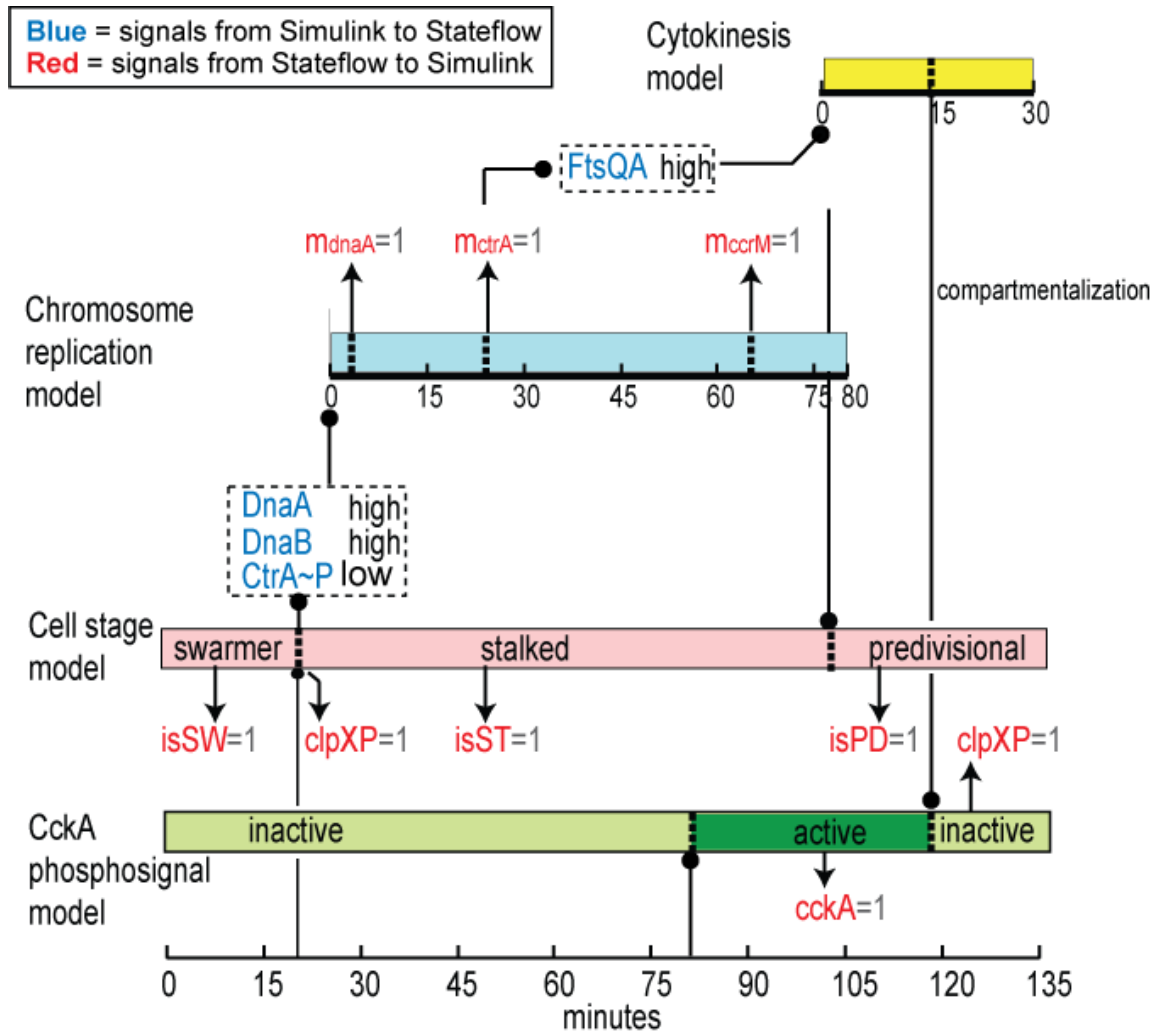


Figure 3.5 A conceptual drawing of the Stateflow subsystem. A pointed arrow with caption indicates that its corresponding binary output variable is switched and fed into the Simulink subsystem. A rounded arrow represents a triggering event for a state transition in one of the four processes. A dashed box represents a condition to be met by the genetic circuit in Simulink to trigger a new process.

Over a cell cycle, Stateflow functions as follows* :

1. The cell is initially in the swarmer cell state (set *isSW* to 1).
2. At 20 minute in simulation time (minSim), the cell transitions from the swarmer stage into the stalked stage (set *isSW* to 0 and set *isST* to 1).
3. During the transition, the CckA phosphosignal is switched to start CtrA proteolysis by the ClpXP machinery (set *clpXP* to 1).
4. When the CtrA level becomes low (through proteolysis) while the DnaA and DnaB levels are still high, chromosome replication is initiated.
5. Chromosome replication is modeled as a linear process lasting 80 minutes. When a fully-methylated promoter is replicated, it becomes hemi-methylated. Starting from the *ori*, the simulated progression of the replication fork reaches the *dnaA*, *ctrA*, and *ccrM* genes (at a time depending upon their position on the chromosome) and successively switches *mdnaA*, *mccrM*, and *mctrA* from 0 to 1 to indicate the promoter is then hemimethylated. Hemi-methylation reduces *dnaA* expression while, it enables *ctrA* and *ccrM* expression.
6. At 80 minSim (60 minSim into chromosome replication), the phosphosignal activates CckA again, switching *cckA* to 1.
7. The Simulink model starts to synthesize CtrA again when *mctrA* is set to 0. When *cckA* is switched to 1, CtrA is phosphorylated into CtrA~P, the active form of CtrA, which activates the *ftsQA* promoter to synthesize FtsQ and FtsA.
8. FtsQ and FtsA are required to start cytokinesis {Martin, 2004 #17}. Cytokinesis is modeled as a linear process that lasts 30 minutes from start of cell constriction to cell separation, with cytoplasmic compartmentalization occurring 18 min {Judd, 2003 #69} before cell separation.
9. The cell cycle stage is changed from stalked to pre-division when cytokinesis starts (set *isPD* to 1 and *isST* to 0).
10. 80 minSim after the replication is initiated, chromosome replication completes and the two chromosomes are separated. *chro_rep* is reset to 0 again.
11. 12 minSim into cytokinesis, the inner membrane of the cell fissions, and the cytoplasm is compartmentalized. Chromosome replication has to complete before compartmentalization

* Specific numbers shown are parameterized in the simulation and could vary depending upon the particular case being studied.

can take place, so the simulation checks if *chro_rep* has been reset to 0 before allowing compartmentalization to happen.

12. Upon compartmentalization, the CckA phosphosignal is blocked in the nascent stalked daughter cell (set *cckA* to 0). The disappearance of the CckA phosphosignal activates CtrA proteolysis by the ClpXP machinery {Iniesta, 2006 #785} (set *clpXP* to 1).

30 minSim after the initiation of cytokinesis, the two daughter cells are separated.

Additional assumptions:

1. No phosphatase signal. There could be a phosphatase signal that works in conjunction with the CckA phosphosignal to accelerate dephosphorylation of CtrA {Iniesta, 2006 #785; Biondi, 2006 #786}. If such a mechanism exists, it would increase the speed and reliability of elimination of CtrA~P.
2. After the initiation of chromosome replication, there is a time window when DnaA is still present but CtrA is not yet re-synthesized. During this interval, we assume that there exists a mechanism to prevent excessive chromosome replication initiation in *C. crescentus*. The mechanism might involve control of the level of activation of DnaA by ATP as with *E. coli* DnaA.

Model parameters

The cell cycle simulation model has a total of 62 parameters. Among them, 29 parameters have experimentally measured values (Table A.2.A, Appendix A), 25 parameters have estimated nominal values, and 9 parameters are used for *in-silico* mutant simulations. The robustness analysis in Chapter 5 found that the cell cycle control circuit design will execute the cell cycle correctly over wide ranges of parameter values.

We use data from {Keiler, 2003 #91} scaled to a 135 min swarmer cell generation time for the timing of the swarmer cell stage, and chromosome replication. Half-lives of DnaA, GcrA, and CtrA have been experimentally determined in swarmer cells and stalked cells {Collier, 2006 #783; Domian, 1997 #290; Gorbatyuk, 2005 #754}, and the active regulation of CtrA proteolysis as a function of the cell cycle has been extensively studied {McGrath, 2006 #859; Iniesta, 2006 #785}. The relatively small dilution effects of cell growth are assumed to be included in the

experimental protein half-life data. Pathways controlling DnaA and GcrA stability have not been characterized, so we modeled the observed dynamic control of DnaA and GcrA stability by setting their half-lives to the reported value at each stage in the cell cycle. Time-resolved measurements of protein phosphorylation states are possible, but with poor resolution, and *in vivo* kinetics of *C. crescentus* phosphorylation reactions are not available. We assume that the phosphosignaling reactions are fast enough that phosphorylation-related switching is much faster than switching by genetic mechanisms or protein degradation. The rationale for choice of all parameter values is in Appendix A.

Even though the estimated parameter values and the postulations made in the model are supported by indirect experimental evidence, there might be alternative explanations for the same observations. In the next chapter, simulation results of the hybrid system model are compared with various experimental measurements to provide another level of validation where the impact of these estimations on the dynamic behaviors of the entire regulation system is evaluated. The robustness analysis described in Chapter 5 takes a step further to show that precise parameter values and detailed characteristics of the individual pathways are not essential to the operation of the *Caulobacter* regulatory system because the modeled feedback control scheme is robust to parameter variations.

Chapter 4

Simulation results

Comparing simulation results to experimental measurements is a crucial step to validate a system model before it is eligible to make predictions, because most reactions included in the model have only been identified by individual experiments. As a synthesis of all known literature, the *in-silico* model provides a unique platform to simulate the interactions between these reactions and assess their influence on the dynamic behavior of the entire system. Even though presented here as sequential steps, validation is actually interwoven into the progressive development of the *Caulobacter* cell cycle model, constantly suggesting clues for improvement. For example, the initial model did not include the methylation control of activation of the *ctrA* P1 promoter; as a consequence of lacking this crucial feedback mechanism, precise parameter values with little tolerance for variation are required for the core engine of the genetic circuit to stay in sync with the lengthy cell cycle processes and it was impossible for the simulation to match the measured protein and mRNA levels. All clues pointed to the seemingly premature activation of CtrA expression, which prompted us to include the methylation control in the model and realize its role for the control scheme, which is to couple the genetic engine to progression of cell processes by expressing the master regulators “just in time” {McAdams, 2003 #58}.

Dynamics of the hybrid system model can be verified by experimental measurements in three fronts: (i) protein level measurable by Western blots; (ii) mRNA level measurable by

Microarrays; (iii) phenomenological events or defective phenotypes observable through microscopy.

4.1 Validation

Protein level

The western blot, also called immunoblot, is a method of detecting protein of interest in a given sample of cell using antibodies binding specifically to the target protein extract {Burnette, 1981 #873}. This method is widely used in the fields of molecular biology, biochemistry and other molecular biology disciplines. Western blot does not provide the exact protein concentration level or the number of proteins molecules in a cell, but computational tools like ImageQuant {GE Healthcare #874} can estimate relative change of protein concentrations during cell cycle progression from a time series of western blot measurements (Figure 2.1B). With the exception of CtrA and CcrM, *in vivo* measurements of the number of protein molecules in the *C. crescentus* cell have not been published. Accordingly, we normalize protein concentrations to the maximum concentration when comparing simulation results to the measurements.

The progression of protein and mRNA concentrations can be followed within the simulation into one or the other of the nascent daughter cell compartments. Figure 4.1A shows predicted concentrations of CtrA, CtrA~P, GcrA, DnaA, and CcrM as a function of time in the cell cycle from the instant of cell separation through cytoplasmic compartmentalization into the stalked daughter cell compartment until cell division. Figure 4.1B follows the protein concentrations into the swarmer daughter cell compartment. The ability to follow predictions of the distinctive molecular concentrations through the cell cycle and into either the swarmer or stalked daughter cell compartments is a unique aspect of this simulation. Experimental observations from synchronized populations (e.g., from Western blots and microarray assays) are averages over many cells, and measurements of samples taken late in the synchronized cell cycle always include signals from both the nascent swarmer and stalked daughter cell compartments. We make the single cell predictions of protein and mRNA levels comparable to observations in synchronized cell populations by (i) averaging the predictions from the swarmer and stalked daughter cell branches of the simulation (Figure 5.1A and Figure 5.2B), and (ii) convolving the

result with a Gaussian distribution with a 5 minute standard deviation to approximate dispersions in cell cycle stage among cells in experimental samples. Figure 4.1C shows the resulting protein level predictions (the curved lines) along with normalized experimental results quantified from Western blots (the circles) using ImageQuant.

mRNA level

By using an array containing many DNA samples, DNA microarray allows scientists to determine the expression levels of thousands of genes within a cell by measuring the amount of mRNA bound to each site on the array {Schena, 1995 #875}. Microarray gene expression assays have been performed on periodic samples from synchronized *Caulobacter* cell populations {McGrath, 2007 #876}. In Appendix B, we compare the mRNA levels from the microarray data to the promoter activation levels modeled by Hill functions in the simulation model. Figure 4.1D shows the temporal profiles of *ctrA* mRNA levels predicted by the Hill function approximation for gene activity. Experimental results from Affymetrix microarray assays of time samples from synchronized cell populations are shown for comparison.

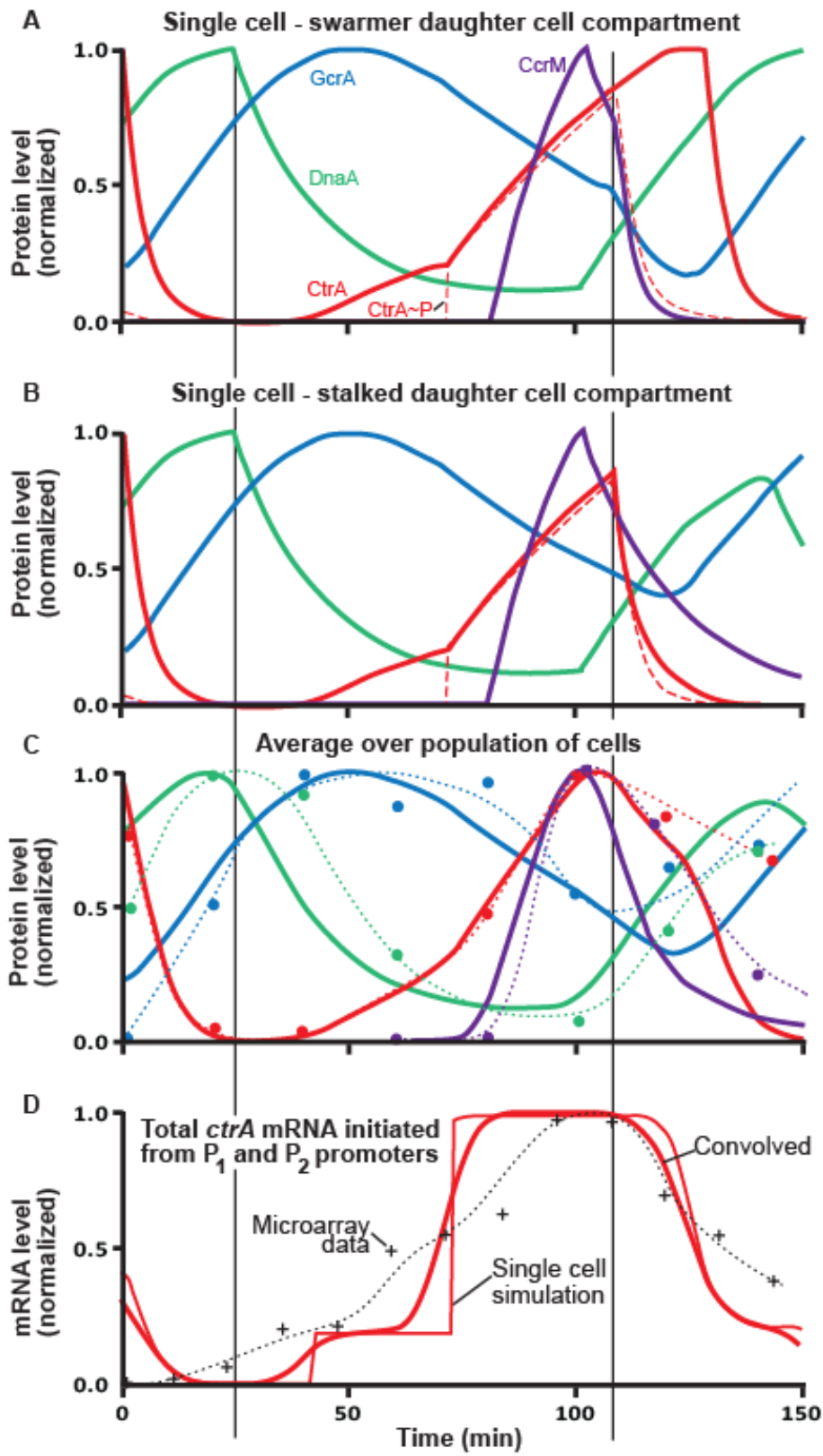


Figure 4.1 Simulation of protein levels (normalized) during cell cycle progression. Panels (A) and (B) show predicted (normalized) levels of the master regulatory proteins tracked into the swarmer and stalked cell compartments, respectively, at the single cell level. After inner membrane compartmentalization at about 117 min, protein concentration levels diverge in the stalked and swarmer daughter cell compartments. (C) Circles: Observed protein levels in synchronized cells (quantified Western blots – Figure 2.1B). (The dotted lines are continuous approximations of the experimental levels.) Curves: Simulated protein levels made comparable with experimental observations by averaging results in (A and (B) and convolving with a Gaussian distribution to approximate random variation around an average in different cell's progression through the cell cycle. The errors in the experimental values are approximately $\pm 10\%$ of the peak value. Loss of synchrony degrades experimental data in predivisional cell. (D) Circles: Observed *ctrA* mRNA levels from Affymetrix microarray assays {McGrath, 2007 #876}.

It is more difficult for the hybridization probes on the DNA microarray to differentiate the activation level of the *ctrA* P1 and P2 promoter due to the noise level present in the experiment. Because the activation of *ctrA* expression is a critical piece in the puzzle, a *lacZ* reporter gene was first genetically inserted into the *Caulobacter* chromosome behind the *ctrA* P1 promoter and then the P2 promoters. These two genetic constructs allow a pulse-label immunoprecipitation assay of Beta-galactosidase (the protein synthesized from the *lacZ* gene) to report the activation level of P1 and P2 separately, which are compared to our simulated levels (Figure 4.2).

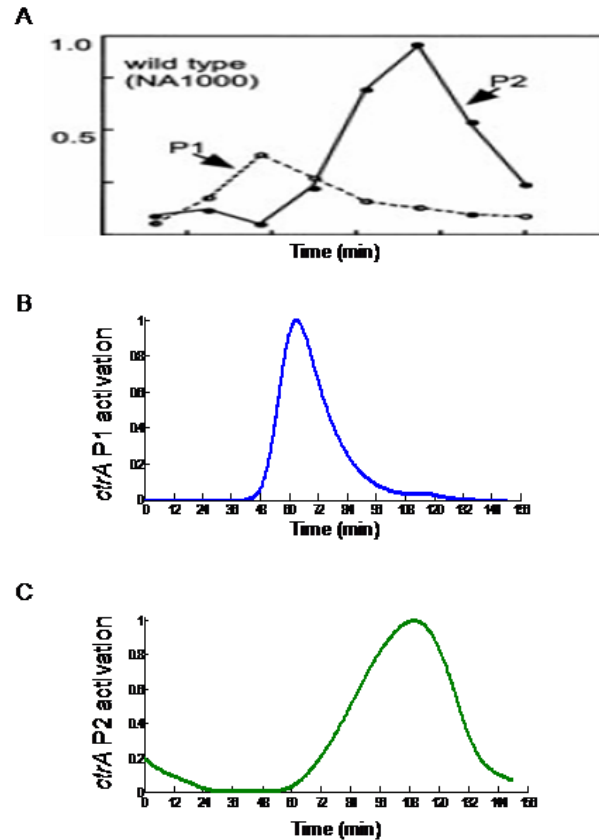


Figure 4.2 The top-left plot shows the expression levels of *ctrA P1* and *ctrA P2* separately during the cell cycle (A) Activation levels of *ctrA P1* and *ctrA P2* promoters measured separately using a pulse-label immunoprecipitation assay of Beta-galactosidase {Reisenauer, 2002 #104}. (B) Activation level of *ctrA P1* predicted by the Hill function approximation in the model. (C) Activation level of *ctrA P2* predicted by the Hill function approximation in the model.

Phenomenological events or defective phenotypes

The discrete part of the hybrid model outputs the timing of the critical cell cycle events of the processive reactions during cell cycle progression. These processive reactions are often initiated when the master proteins levels cross certain thresholds and when these reactions reach certain critical stages, they conversely affect the expression level (mRNA) level of involved genes through methylation and the phospho-pathway. Figure 4.3 displays both the continuous protein patterns and the discrete stages of the cell processes simulated by the hybrid system model, demonstrating the operation of the cell cycle feedback control. The timing of the discrete events have been confirmed by microscopic images from various literature.

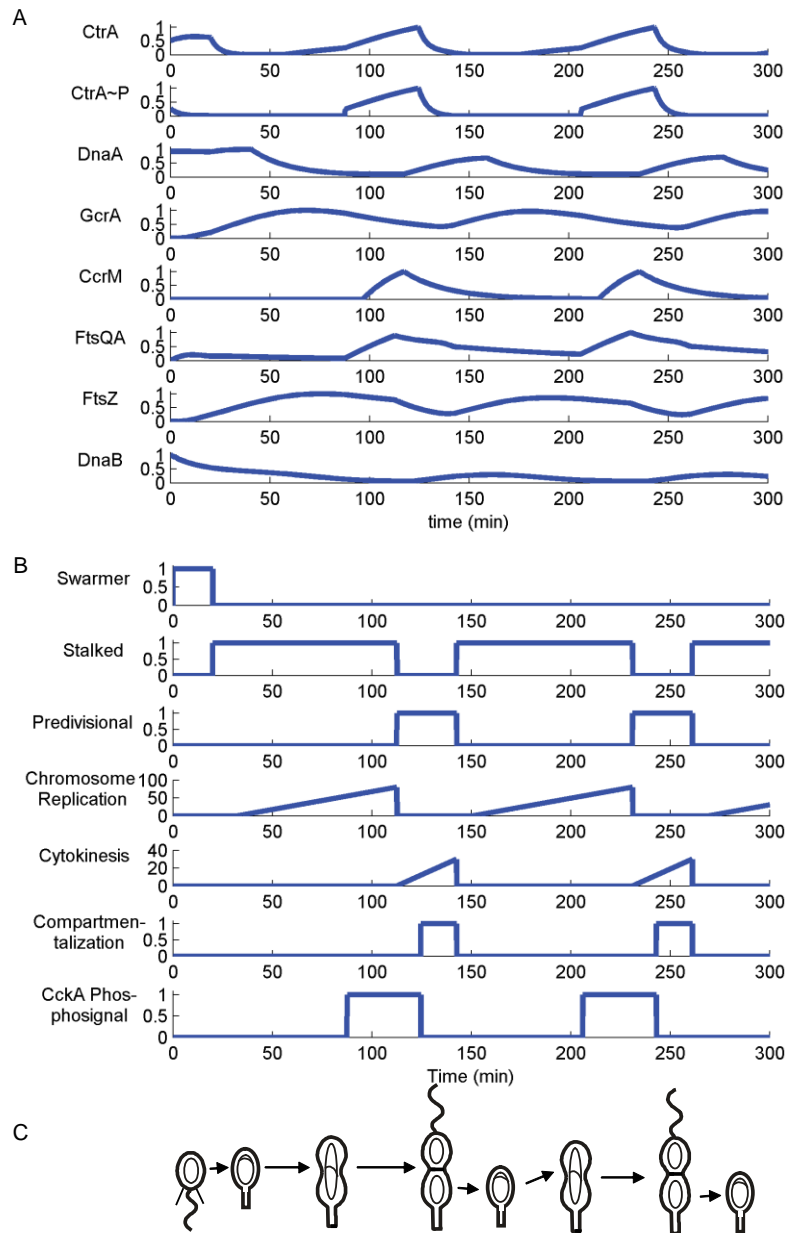


Figure 4.3 Simulated dynamics of the hybrid system model tracking into the stalked compartment (A) Protein concentration from the continuous ODE models in Simulink (B) State variables from the discrete phenomenological model in Stateflow (C) A cartoon showing cell cycle progression corresponding to (A) and (B)

4.2 *In silico* mutant strain

Shown in the previous section and Appendix B, patterns of protein and mRNA concentrations predicted by the wild-type strain simulation using nominal kinetic parameter values characteristic of cells in log growth phase (Appendix A) agree with various measurements within experimental error, particularly after recognizing loss of synchrony in the predivisional cell data. To further validate the hybrid control simulation of the *in vivo* cell cycle control system, we created *in silico* mutant strain simulations that emulate several laboratory *C. crescentus* mutant strains. Besides validation of the model, another objective of these comparisons was to understand the reasons for the experimentally observed phenotypes in greater depth. Additional details of these simulations including graphs of the *in silico* simulation datasets are online at <http://www.stanford.edu/group/caulobacter/CellModel>.

Table 4.1 shows the four mutants that were simulated and the changes that were made to the wild-type model to create the mutant simulation. In each case, the simulation predicts (i) the concentration profile of each protein in the model in single cells as a function of cell cycle time when followed into either the swarmer or the stalked compartment of the predivisional cell, (ii) whether the cell can progress through each stage of the cell cycle, and (iii) whether DNA replication and cytokinesis occur normally.

Table 4.1: mutant phenotypes

Characteristics of mutant strains	Genotypes of mutant strains	Refs	Phenotypes <i>in vivo</i>	Parameters changed from wild-type parameters for mutant simulations
Strain where GcrA can be depleted (LS3707)	CB15N $\Delta gcrA$ $P_{xyl}::gcrA$	{Holt zendo rff, 2004 #33}	The cell cycle is arrested at the stalked cell stage, and cells finally die in the absence of GcrA.	Maximum GcrA synthesis rate from the <i>gcrA</i> promoter $pgcrA=0nM/s$
Strain that accumulates CcrM constitutively (LS1)	CB15N $PlacZ::ccrM$	{Zwe iger, 1994 #373}	Cells are slightly elongated and accumulate supplementary copies of the chromosome.	CcrM synthesis rate from a constitutive promoter added in the model $pccrMoe=100nM/s$
Strain that can accumulate stable and constitutively active mutant CtrA proteins	CB15N $pXylX::$ $ctrAD51E\Delta3\Omega$	{Dom ian, 1997 #290}	Cells do not initiate DNA replication and do not divide. Cells elongate before dying.	Binary switch controlling the phosphorylation state of CtrA and the protein half-life of CtrA under active proteolysis by ClpX

				isAlwaysCtrAP=1 hlCtrAf=200 min
Strain where the <i>ctrA</i> gene is moved to a position next to the terminus of replication of the chromosome (LS3355)	CB15N <i>ctrA</i> Δ2::pAR3 58	{Reis enaue r, 2002 #104}	Cell size is sometimes irregular	Relative location of the <i>ctrA</i> gene on the chromosome zpctrA=1

The simulation predictions for all cases checked were consistent with the *in vivo* phenotypes. Simulation results and their relation to the mutant strains in Table 4.1 are as follows:

GcrA depletion strain: The simulation predicts that CtrA~P will not re-accumulate after the stalked cell stage, so FtsQA does not accumulate enough to initiate cytokinesis. As a result, the cell cycle arrests at the stalked cell stage in the simulation (Figure 4.4). The simulated levels of DnaA and CtrA suggest that DNA replication may still happen in these cells before cell death.

Strain with constitutive accumulation of CcrM: The simulation predicts that the re-accumulation of CtrA in pre-divisional cells will be delayed ~20 minutes, while DnaA will accumulate at high concentrations throughout the cell cycle. This suggests that over-initiation of DNA replication may take place in these mutant cells as is observed experimentally. As a consequence, the re-accumulation of FtsQA and cytokinesis will be delayed.

Strain with constitutive accumulation of CtrA~P: The simulation predicts that accumulating CtrA~P will block the initiation of DNA replication. Since cytokinesis is blocked when DNA replication is blocked, the cells will arrest after the stalked cell stage as observed *in vivo*.

Strain with the ctrA gene moved next to the DNA replication terminus: The simulation predicts that CtrA re-accumulation in predivisional cells will be delayed by ~15 minutes as is observed. As a consequence, the synthesis of FtsQA and cytokinesis will also be delayed, so that the cell cycle will be slightly longer than for wild-type cells. The consistency between the predictions from simulation of the *in silico* mutants and the *in vivo* phenotypes provides additional evidence that our model corresponds to the biological cell cycle control circuitry. Moreover, the predictions from *in silico* mutant simulations provide quantitative insights into how the cell cycle is affected by a given mutation.

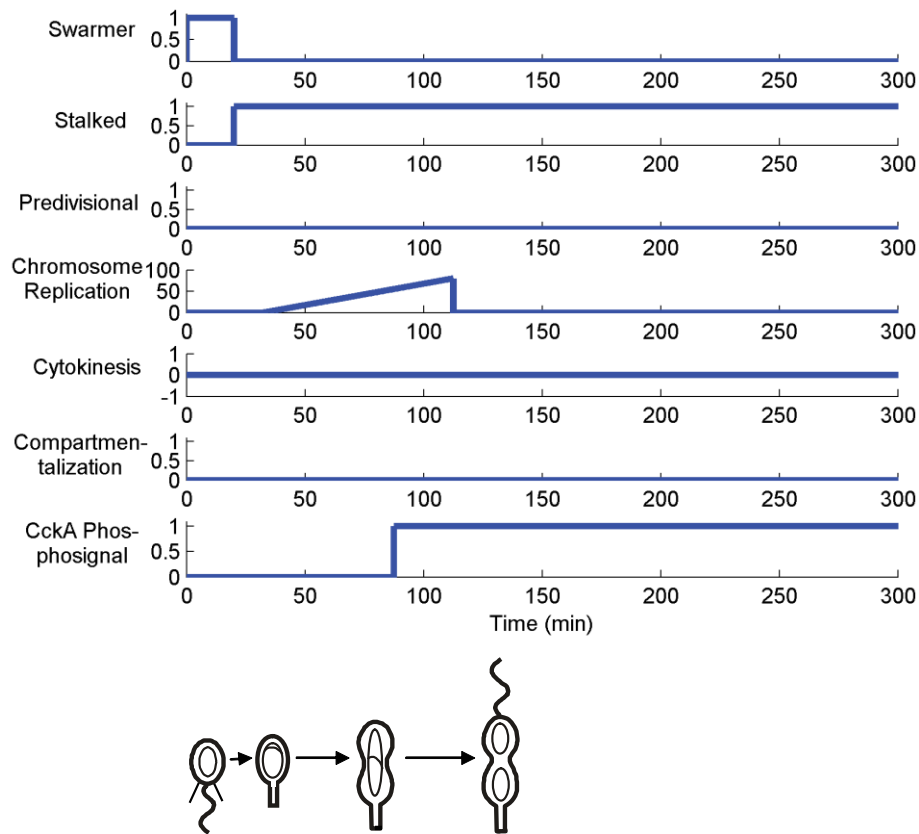


Figure 4.4 *In silico* GcrA depletion mutant are arrest as stalked cells. The top part shows the state variables from Stateflow. The defective phenotype is drawn intuitively at the bottom.

The simulations were performed using the Matlab-based simulation of the wild-type *C. crescentus* cell cycle control system. We used the same differential equations, parameter values, and initial conditions as for wild-type cells, except for those parameters that were changed to simulate a mutation of interest. Time varying intracellular concentration levels are predicted for the eight different proteins included in the model.

The consistency of the simulation results with these experimental observations illustrated that our hybrid simulation model is capable of capturing the dynamic behaviors of the *in vivo* cell cycle regulatory circuit. The next chapter analyzes the robustness of the cell cycle control and shows that the cell cycle will operate correctly even if there is wide variation in kinetics of reaction rates in different pathways.

Chapter 5

Robustness

The cell cycle control system of the organism has to be reliably competitive over the entire range of potential environmental circumstances, or the organism will not survive. A relatively more robust organism that can survive a bit wider temperature range or a bit longer period of starvation has a major evolutionary advantage. Biological regulatory systems often possess mechanisms for robustness that are only expressed under unusual or stress conditions to make the regulatory system robust, or insensitive to environmental changes, genetic mutations, stochastic fluctuations, and noise {de Visser, 2003 #797; Goulian, 2004 #801; Kitano, 2004 #807; Kitano, 2007 #808; Rosenfeld, 2005 #811; Thatcher, 1998 #818}, but models of regulatory systems are usually based on observations and measurements made under typical laboratory conditions, making them incapable of investigating these contingent mechanisms. For example, 73% of the 6000 genes in the budding yeast *Saccharomyces cerevisiae* have shown to be nonessential by gene depletion mutations {Dwight, 2004 #851; Giaever, 2002 #852}. These nonessential genes are likely to be part of pathways that play an essential role in diverse habitats when facing environmental or genetic perturbations {Thatcher, 1998 #818}, but their roles will be dismissed in models built solely on data collected under regular lab conditions. Therefore it is always challenging for biologists to thoroughly evaluate the robustness of a regulatory system. This chapter presents a novel approach to exploring cell cycle robustness by leveraging a system verification methodology from digital circuit design. In addition to proving that the *Caulobacter*

cell cycle is extremely robust, this method revealed nuances in the regulation scheme that contribute to the robustness under unusual circumstances.

5.1 Robustness analysis on system model

It has been proposed that robustness can be used as a criterion to test if a model of a regulatory network is realistic {Morohashi, 2002 #810; Savageau, 1971 #812}, because a model that lacks tolerance for parameter variation suggests that important pieces of the regulatory system is missing or modeled incorrectly, otherwise the corresponding organism will not be able to survive in the wild. Furthermore, checking how a model behaves over the entire parameter space may identify robustness-enhancing mechanisms not expressed under laboratory conditions as well as insights on the structure of the modeled regulatory network.

Measurements in vivo shows that a synchronized population of *Caulobacter* cells originated from the same strain and living in the same controlled lab environment becomes increasingly unsynchronized during cell cycle progression, confirming that the cell cycle regulation has to tolerate various speed of cell cycle progression, probably due to environment (local crowdedness) and minor genetic mutations accumulated after several generations (Figure 5.1) {Judd, 2003 #69}. The cell cycle time of *Caulobacter* cells living in different media in the lab could vary significantly, e.g. 90 minutes in PYE whereas 150 minutes in M2G, which indicates the influence of nutrient levels on cell processes and protein synthesis. In their natural habitat with much more diluted media, *Caulobacter* cells are subject to far greater perturbations and starvations, thus to be a competitive species, the regulatory system has to be robust; that is to always copy the chromosomes and divide the cells successfully regardless of the duration of cell cycle.

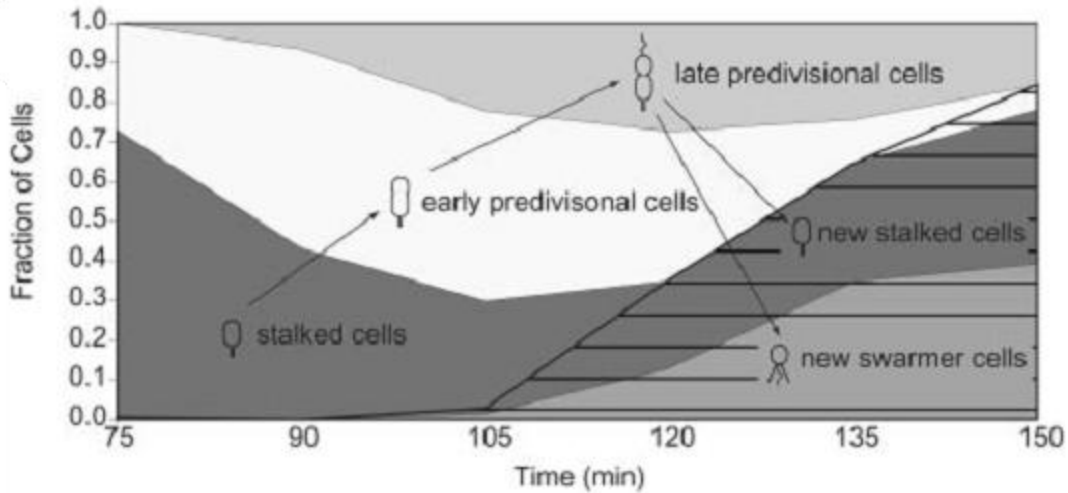


Figure 5.1 Measurements illustrating the loss of synchrony during cell cycle progression. Fraction of swarmer cells, stalked cells, and early and late predivisional cells present in the synchronized population as a function of time. Cell type was determined by morphology. Seventy-five minutes into the cell cycle, the population consists primarily of stalked cells. As the cell cycle progresses, stalked cells develop into predivisional cells, which then divide to form new swarmer and stalked cells (hatching) {Judd, 2003 #69}.

Robustness has been predominantly analyzed by exploring parameter sensitivities based on the premise that a robust model should be insensitive to the precise values of its parameters {Alon, 1999 #789; Barkai, 1997 #794; Ingolia, 2004 #805; Stelling, 2004 #816; von Dassow, 2000 #822; Yi, 2000 #827}. We performed a limited parameter sensitivity analysis on a subset of the parameters in the simulation, and the results suggested that the operation of the cell cycle control is robust. However, parameter sensitivity analysis and stochastic simulations like Monte Carlo {Fishman, 1995 #877} do have their limitations: (i) It is often computationally intensive to exhaustively search the entire parameter space to identify all failure cases. (ii) It is often not trivial to interpret the results: why specific sets of parameter values can cause a regulatory circuit to malfunction? (iii) Parameter sensitivity analysis seldom provides clues to robustness-enhancing mechanisms not yet included in the model. Even though the parameter sensitivity analysis provided an initial assessment of the robustness of cell cycle regulation, it fell short as a diagnostic tool for analyzing the robustness of the model under all possible perturbations faced by *Caulobacter* cells in their natural environment.

In contrast, when testing an engineering model, engineers usually adopt a “top-down” view of the system from a functional perspective rather than a “bottom-up” view with individual parameters. Specialized tools have been developed to exhaustively identify hazardous conditions in engineered systems and follow up with potential solutions. One set of tools are called symbolic model checking, which has been used to analyze robustness of asynchronous circuits, circuits without a clock signal to synchronized all the signals {Clarke, 1986 #878; Batt, 2007 #879; Burch, 1990 #887; McMillan, 1992 #889}. As a naturally evolved asynchronous system “engineered” with biological rather electrical components, can *Caulobacter* cell cycle regulation be analyzed by symbolic model checking as well? To answer this question, we have to examine the operational principles of electrical engineering system applicable to symbolic model checking and find out if they are still valid for *Caulobacter* cell cycle regulation.

5.2 Discrete abstraction for model checking

Model checking generally refers to verifying whether a structure is a model of a given logical formula. Methods have been developed to algorithmically verify formal systems. This is achieved by verifying if the structure, often derived from a hardware or software design, satisfies a formal specification, typically a temporal logic formula {Clarke, 1981 #884; Queille, 1982 #885; Emerson, 1985 #896}. Symbolic model checking makes a discrete representation of the system structure using a formula in propositional logic, often in some forms of binary decision diagrams (BDDs) {Hu, 1993 #886; Burch, 1992 #888; McMillan, 1992 #889}.

How does symbolic model checking manage to identify hazard cases in engineering systems without sweeping through all parameters? Instead of characterizing the exact shape of a signal which is heavily influenced by many parameters, the level of the signal is divided into discrete regions corresponding to its functional role. For example, a signal level is considered “high” when it is above the activation threshold of its downstream devices, and “low” vice versa (Figure 5.2A, Figure 5.2B). Progressive processes are also divided into discrete states and treated as state machines, which is exactly how we treated the phenomenological models in our hybrid system representation of cell cycle regulation.

Different from some previous work using Boolean network which treat ongoing reactions as instantaneous and simulate them in synchronized steps, the dynamics of the signals are captured by introducing the notion of timing and delay in the system abstraction for symbolic model checking (Figure 5.2C, Figure 5.2D). Timing refers to the time instant when an event occurs, i.e. when a signal crosses a threshold or a state machine transitions into the next state. The concept of delay refers to the response time for a downstream output signal to cross its threshold after its input signal crosses the activation threshold (Figure 5.2D) or the transition time for a state machine to enter the next state after the initiating condition is satisfied.

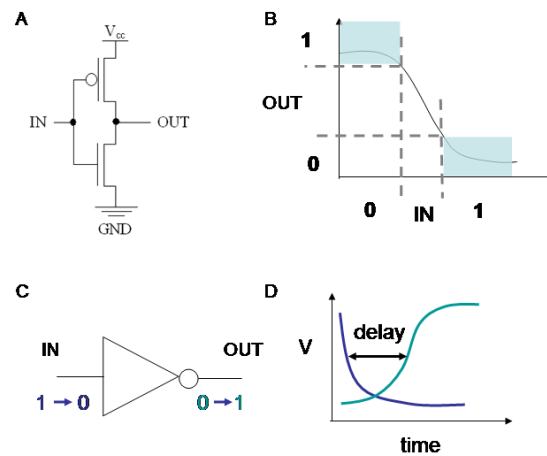


Figure 5.2 Discrete abstraction of electronic signals. (A) A schematic of a CMOS inverter which outputs an inverse signal. (B) A steady state transfer function curve with output level plotted against the input level. The exact shape of the curve depends on tens or hundreds of device parameters. The blue shaded areas are “flat” saturation regions that could be abstracted into two discrete regions: “high” and “low”. The dashed lines mark the threshold region between the discrete regions, in which neither the input nor the output could be defined as high or low. An analog device such as an amplifier operates in this range. (C) A transition of the input causes a transition of the output. (D) The dynamic response characteristics of the inverter could be captured by a delay variable which marks the time lapsed between the input and the output level crossing the threshold regions.

This discrete abstraction of signals and processes allow algorithms like symbolic model checking to investigate effects of parameter variation by exploring timing and delay variation instead of inspecting individual device parameters {Hu, 1993 #886; Burch, 1990 #887; Burch, 1992 #888; McMillan, 1992 #889}. The effects of parameter variation on the steady state transfer functions are captured by varying threshold levels and the effects on the dynamic

behavior are captured by varying delays. Furthermore, variation of threshold levels ultimately translates into delay variation as well because an altered threshold level means that now it takes the same signal less or more time to cross it, with the extreme case where the threshold level is changed so much that it exceeds the maximum or minimum signal level so the delay becomes infinite.

A caveat of this abstraction method is that the discrete representation preserves all properties of interest only when the operation of the original continuous system adhere to certain principles: (i) The steady state transfer function curve of every signal has a sigmoidal shape (Figure 5.2B) so that the noise or signal value uncertainty gets attenuated when devices are cascaded (in other words, noise does not become amplified during cascading) (ii) If we freeze the inputs and wait long enough, in the steady state all signals and processes end up residing in one of their discrete states, even though the transitions can take a long time to reach this state. When these two conditions are met, a system is considered to be a discrete or “digital” system which can be safely abstracted for symbolic model checking without risking losing properties of interest.

The explanation of the first principle needs a little elaboration. The sigmoidal shape of the transfer function has two saturation regions separated by a threshold region with a slope (derivative) bigger than 1, therefore when an input signal falls into the threshold region due to noise or signal degradation, the output signal gets “pushed” back out into the discrete (saturation) regions, thus preserving the integrity of the signal. This is essential for the discrete abstraction because otherwise if the slope is less than 1, in a cascade, signals keeps getting degraded and eventually falls into the threshold region, making “1” or “0” meaningless. In such a system, noise or stochastic variation gets amplified until all assumptions on a digital system are violated.

In the discrete representation, the finite number of system states owing to the discrete state variables makes it computationally feasible to exhaustively search the entire state space for potential failures in the operation when parameters vary }{Alur, 2000 #880; Casagrande, 2007 #881; Batt, 2007 #879; Mysore, 2007 #882; Lincoln, 2004 #883}. Moreover, the transformation of the originally continuous state space into a discrete one makes it unnecessary to examine detailed dynamic characteristics of the signals; rather what only matters is if and when a signal crosses its threshold or a state machine transitions into the next state. In other words, by checking

the timing and causality of the critical events rather than the exact shape of signal curves, the tool is able to abstract the model from the functional perspective and examine the collective effects of all parameters on the final outcome. If the timing variation of a critical event exceeds certain limits, either happening too early or too late, its downstream events could be disrupted and the intended actions might never take place properly.

One common case that causes unwanted outcomes in an electrical circuit is race (Figure 5.3) {McClusky, 1986 #809}. When multiple signal chains (pathways) lead to inputs of a logic gate to make a decision, parameter variation affect the timing delay of each signal chain and thus the order of the arrivals of these signals at the inputs, which can generate unexpected intermediate outcomes called glitches. In a combinational logic circuit with no feedback, glitches do not cause malfunctions since the signals always settle into the correct final values. However, if there exist feedback loops in the circuit, the glitches might change the final outcome of the feedback loops, causing the operation to wander into unwanted states and eventually fail. A well designed circuit should deploy extra circuitry to prevent these hazards (failures) from occurring under parameter variations. Symbolic model checking has been used by design engineers to identify all potential timing hazards in the design and report the step-by-step signal sequence leading to a hazard.

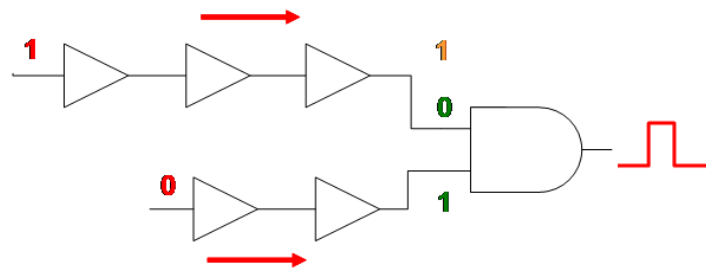


Figure 5.3 Race and glitch in electrical circuits. The green digits are current state of the inputs into the AND gate. The red digits are upcoming signals passing down through the two signal chains. The top chain is usually slower owing to the extra buffer, but parameter variation could result in an early arrival of its signal (the orange digit), causing the AND gate to output 1 temporarily before settling back to 0, forming a glitch (shown in red).

We postulated that *Caulobacter* cell cycle regulation is a robust system and our preliminary parameter sensitivity analysis has shown that the hybrid system model was somewhat insensitive to parameter variations. Still, many questions remained unanswered: i) how robust the cell cycle regulation is (how many potential failures cases can be caused by parameter variations)? ii) What

are the exact conditions or deviation of parameter values that lead to incorrect replication and division in our model? iii) Can these conditions provide clues to *in vivo* mechanisms not yet included in our simulation model? iv) Do the non-essential mechanisms in the simulation model contribute to robustness in any way? To answer these questions using symbolic model checking, we first need to check that the *Caulobacter* cell cycle regulation system do adhere to the operational principles of the engineering systems that makes it eligible for discrete abstraction with timing.

5.3 Discrete representation of *Caulobacter* cell cycle regulation

Perturbations introduced *in vivo* or parameter variation *in silico* can affect a cell signal (regulatory protein) by changing its level or timing (Figure 5.4A). For example, the *Caulobacter* cell cycle accommodates various rates of progression caused by nutrient availability, temperature fluctuation, and ligand binding affinity due to genetic mutations. In the case of protein synthesis, each one of the various parameters could affect the protein synthesis and degradation rates, generating different characteristic curves. From Chapter 3, we know that protein synthesis and degradation corresponding to a repressive promoter can be approximated by the following approximation:

$$\frac{d[B]}{dt} = \frac{\beta}{1 + \left(\frac{[A]}{[C_d]}\right)^n} - \lambda \cdot [C] \quad (5.1)$$

where [A] is the level of the repressive transcriptional factor and [B] is the level of accumulated protein. Variation of the synthesis or degradation rate results in different accumulation rates of the targeted protein B, thus after the level of its repressor A goes down, it takes different amount of response time, or delay, for protein regulator B to reach the threshold level of affecting downstream promoters (Fig. 5A). Stochastic variation or noise of the threshold level also leads to

difference in the delay. Hence the dynamic property of this repressive promoter with its corresponding accumulation of protein B can be captured by using the concept of delay without losing properties of interest as far as transcriptional regulation is concerned.

The steady state transfer function of [B] vs. [A] is achieved by setting $\frac{d[B]}{dt} = 0$:

$$\text{Steady State: } [B] = \frac{\beta / \lambda}{1 + \left(\frac{[A]}{[C_T]}\right)^n} \quad (5.2)$$

The steady state transfer function curve between the output level and the input level shows a sigmoidal shape due to the hill function approximation with cooperativity $n > 1$ (Figure 5.4B). The two saturation regions separated by a transition region allows us to practice the same exercise as we did to digital engineering systems and divide the operating space into two discrete regions separated by a threshold region. When the output protein level is above the threshold region, it is statistically significant enough to activate or repress downstream promoters, and when the level is below, it is statistically insignificant for influencing downstream promoter. If the output level is within the threshold region region (the steep part of the S shaped curve), or in other words, right on the threshold, the process of regulating downstream promoters become stochastic and subject to probability, which is obviously undesirable for the regulatory systems that tries to produce a deterministic outcome.

The higher the cooperativity n is, the steeper the threshold regions is, which suppresses noise propagation and increases noise margin when the signals are cascaded (Figure 5.4C). This might explain why high cooperativity such as dimers and trimers are so common in biological reactions, especially in regulatory systems, because they form more noise resistant, and therefore, better devices for signal propagation and performing logic. As long as the input signal of a cascade is within the discrete operation region (either “low” or “high”), it is guaranteed that the signals along the cascade leading to the final outcome will all reside in the discrete regions when the final steady state is reached, without the risk of getting stuck in the threshold region.

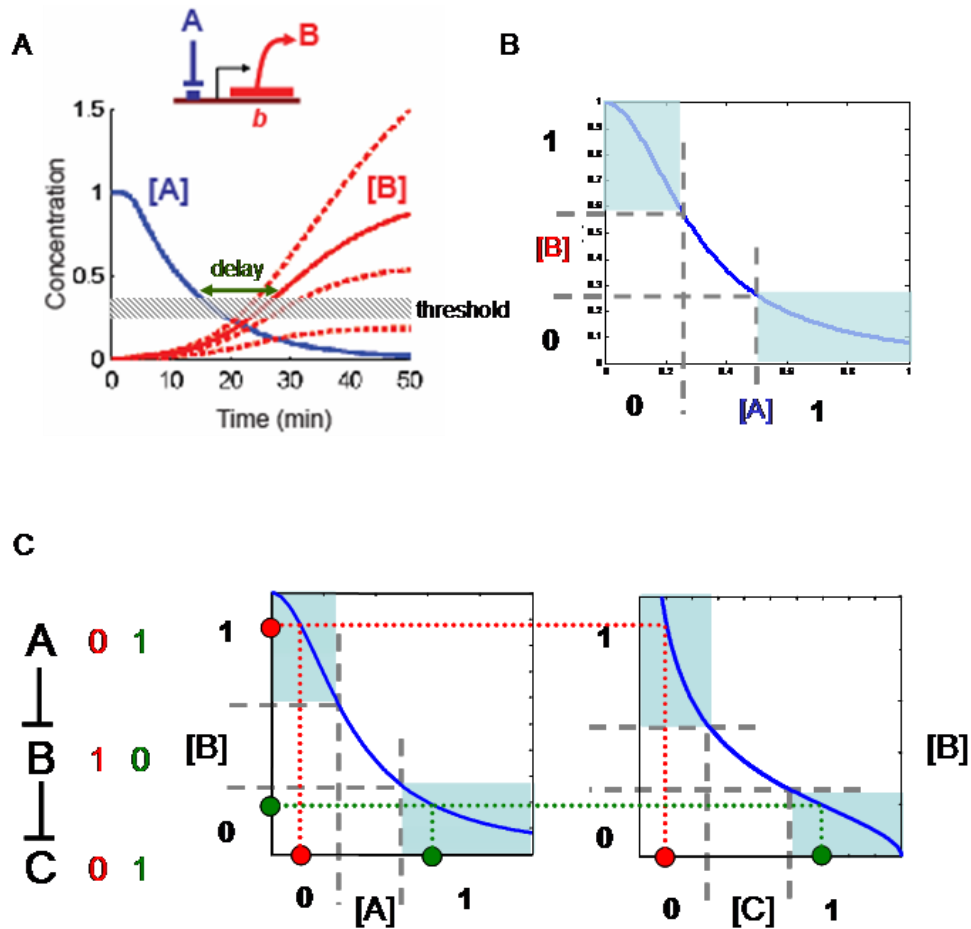


Figure 5.4 Discrete abstraction of transcriptional regulation. (A) An illustration of effects of parameter variation on the dynamic response of a regulator. Different accumulation rates of regulator B due to parameter variation causes deviation from the normal delay it takes regulator B to cross the threshold for affecting downstream regulators. (B) A steady state transfer function curve plotting [B] against [A]. Similar to Figure 5.2B, the operation region can also be divided into two discrete regions (shaded by blue) separated by a threshold region (separated out by the dashed lines). The steepness of the threshold region is decided by the cooperativity of the A binding site on the b promoter, a coefficient in Hill approximation. (C) In a cascade of regulators, discrete abstraction stays valid because the shape of the Hill function keeps the noise margin and preserves the discrete regions. The level of the regulator A results in two outcomes from the cascade (shown in red and green digits on the left), and the steady state transfer functions guarantee that the two operating points (circles and dashed lines in red and green respectively) stay in the discrete regions and never slip into the threshold regions.

The outcome of the *Caulobacter* cell cycle regulation is discrete and deterministic – replicate the chromosome and divide the cell. The cell cycle cannot have a partial product such as 70% completion for 70% of regular expression levels. *In vivo* measurements have indicated that the promoters for the master regulator genes are much stronger than the minimum level and there are regulated proteolysis and/or phosphorylation to disable master regulators to ensure that the levels of the master regulators will never reside in the threshold regions in steady states {Quon, 1996 #329; Judd, 2003 #69}. Therefore, the regulatory network does follow the guidelines for a logic circuit, even though the time scale of this biological circuit is much longer than a typical electrical circuit.

Our examination has shown that the *Caulobacter* cell cycle regulation indeed adheres to the two operating principles required for valid discrete abstraction without loss of properties of interest. These unique properties of the *Caulobacter* cell cycle regulation allows us to make an equivalent asynchronous digital circuit representation of the control circuit¹, making it compatible to robustness analysis using symbolic model checking.

5.4 Timing verification

This section describes the robustness analysis of the *C. crescentus* cell cycle control circuitry using symbolic model checking. First, the concept of timing analysis of the *C. crescentus* cell cycle as a finite state system is described. Then we describe conceptually how the converted cell cycle model represented by discrete state variables is analyzed by NuSMV {Clarke, 1986 #878; Cimatti, 2002 #890}, a tool for symbolic model checking. As stated previously, timing analysis has long been used in engineering to check the robustness of electrical circuits, and to determine if the circuit will always generate the correct logic outcome regardless of its environment and

¹ The applicability of symbolic model checking to cell cycle regulation cannot be taken for granted regarding other biological systems. For instance, metabolic networks have been shown to optimize its fluxes to yield maximum growth, a quantity rather than a discrete decision. Therefore, the entire metabolic network cannot be abstracted into a discrete representation, even though some of its enzymatic control might still behave as digital circuits. Another example is the phage λ lysis-lysogeny decision circuit, in which fluctuations in rates of gene expression lead to non-deterministic bifurcation helped by positive feedback {Arkin, 1998 #793; McAdams, 1997 #843}.

random variation in parameter values. NuSMV was used to check the robustness of the *C. crescentus* cell cycle control circuit to identify hazardous conditions, that is, potential situations where the circuit might not complete the cell cycle.

We based the equivalent asynchronous digital circuit representation of the cell cycle regulation on the validated hybrid system simulation model described in Chapter 3, which allows the symbolic model checking software to exhaustively search the finite state space for potential failures brought upon by parameter variations. Figure 5.5 illustrates the rationale underlying our approach to deriving the discrete abstraction from the hybrid simulation model and application of the model checking methodology for robustness analysis. The figure shows eight events in an interval centered on the time of daughter cell separation that will occur in a stalked cell cycle and affect operation of the control circuit. The events occur at times labeled by the T_i 's. T_1 , for example, is the time of cytoplasmic compartmentalization that interrupts the CckA/ChpT phosphosignal and precipitates elimination of CtrA~P. The specific timing of events and the intervals between the events will depend on the kinetic parameters of the system, whether *in vivo* or *in silico*. (Owing to stochastic variation in reaction rates from cell to cell in a population, actual reaction rates in different cells will vary and consequently cells will show dispersion in the rate of progression through the cell cycle.) Thus, the rate of production of any protein X will depend on the kinetic parameters of its transcription and translation, and the time for X to activate a dependent process will depend on the kinetics of its binding at a target site that determine the concentration range for action. The rate of production of X will also depend on availability of substrates (e.g., amino acids) that will vary with environmental nutrient levels. There is a minimum average cell generation time under optimal high nutrient conditions. Under either carbon or nitrogen starvation, the cell cycle comes to a controlled stop (see below), suggesting that there is also a maximum to the sustainable generation time. Thus, a defining element of the robustness of the *C. crescentus* cell cycle control is its ability to achieve reliably correct ordering of cell cycle functions between these extremes of cell growth rate. Our method for exploring this using symbolic model checking is to consider variation of the event times (i.e., the T_i 's in Figure 5.5) directly, rather than individual variation of the many kinetic parameters that determine the times. This approach to robustness analysis is more computationally efficient and complete than using a Monte Carlo method to explore the parameter space.

For example, consider the particular situation shown in Figure 5.5, in which the condition for activation of DNA replication involving the CtrA, DnaA, and DnaB proteins is satisfied at T3. After a delay, the *dnaA* gene is duplicated at T4. Consequently, its promoter region becomes hemimethylated and the rate of *dnaA* transcription is greatly reduced {Collier, 2007 #784}. Although there are essential ordering relationships between some timing events, for example: $T2 > T1$, $T4 > T3$, $T6 > T5$, $T6 > T4$, and $T8 > T7$, the lengths of the various inter-event intervals, or delays, are determined by the specific reaction rates that occur by chance in each cell under the prevailing environmental conditions.

To facilitate exploration of the effects of timing variations, we simplify the protein concentration profiles as shown in Figure 5.5 so that they are either "low" (under the thresholds of downstream sites of action) or "high" (above the thresholds of downstream sites of action) because as described previously, other than time and direction of traversing the threshold region, details of the temporal profile of these variables are not significant to the operation of the circuit. This yields a discrete logic abstraction of the signaling and the time points that identify the timing of the transitions between the two states (e.g., T5 and T6 in Fig. 3A).

Other events can also be characterized as binary signals or finite state machines with discrete stages. For example in Figure 5.5, the cell changes from not-compartmentalized to compartmentalized at T1, and the *dnaA* promoter changes from methylated to hemimethylated at T4. This simplification of the cell cycle control circuit by using discrete signal levels with discrete transition timing produces an abstraction of the biological circuit equivalent to an asynchronous sequential digital circuit – a network of logic elements and simple state machines with variable delays. Within this abstraction, it is only necessary to examine each distinctive ordering of events to determine whether the circuit will function correctly. Following electrical circuit analysis procedures for similar problems, we use a software model checking tool to search the immense space of all feasible orderings of the T_i 's for cases where the circuit might fail. This approach using a symbolic model checker {Clarke, 1986 #878; Burch, 1990 #796; Burch, 1990 #887} is used to check for correct operation of concurrent systems such as electrical circuits and network protocols; we use it to check the cell cycle control circuit, which is also a concurrent system. The model checker we used, NuSMV {Cimatti, 2002 #890}, takes as input the logic description of the *C. crescentus* cell cycle model (created from the validated simulation model as

described above) and a specification of the limitations on ordering of transitions that must apply for a viable cell. The program checks all feasible orderings of the transition times and notifies the user of any event timing “hazards” where the specification would be violated. A detailed discussion on the implementation of the cell cycle regulation model compatible to NuSMV can be found in Appendix D and online scripts at <http://www.stanford.edu/group/caulobacter/CellModel>.

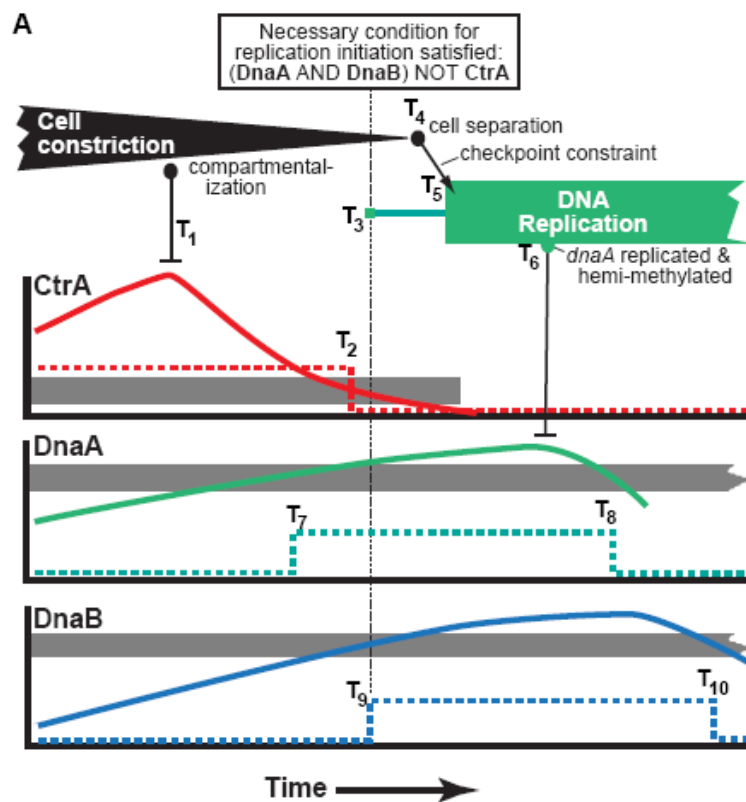


Figure 5.5 Stalked cell cycle events affecting control circuit operation in the interval around cell separation. Graphs show protein concentration patterns. Gray bands: “Threshold” ranges relating to initiation of DNA replication. [CtrA] above the band represses initiation; below, it does not. The dotted CtrA line illustrates the binary signal abstraction used in the robustness analysis. Thus, T2 approximates the time when [CtrA] transitions from repressing to not-repressing initiation. Other T_i 's are times of other events. The robustness analysis examined the effect of all patterns of T_i orderings on cell cycle control.

In the NuSMV input language, a system is described as a collection of state variables ranging over finite sets of discrete values representing states, along with rules for updating these

variables as the system progresses through different states. For instance, a state variable is assigned to represent the state of chromosome replication, which can take on different discrete values indicating pre-replication, initiation of replication, and the middle of replication. A progression of states of the system through time is called a path and the system may be able to progress through many possible paths. In NuSMV, a current state may have many alternative successor states.

A strict mathematical discussion of the model checking algorithms used by NuSMV is beyond the scope of this thesis {Cimatti, 2002 #890}. However, Figure 5.6 intuitively illustrates how model checking extracts the timing information from the discrete representation of the cell cycle and exhaustively examines the possible ordering of events to identify failure cases. As shown in Figure 5.6A, at time T_0 , the chromosome replication fork passes through and hemimethylates the *dnaA* promoter, causing the DnaA level to drop below its threshold at a later time T_2 . Meanwhile, GcrA is being synthesized and its level will rise above its threshold at time T_1 . The replication fork continues to advance after T_0 and will pass the *ctrA* promoter at T_3 . In a normal wild-type simulation with nominal parameter values, T_2 happens before T_1 and T_3 , but when parameter variation is taken into consideration, any one of the three events at T_1 , T_2 , and T_3 could happen first after T_0 . To explore all three timing scenarios, which cover every possible parameter variation, symbolic model checking essentially constructs a tree graph in the discrete state space (Figure 5.6B). From the root vertex corresponding to the discrete state variables at T_0 , three separate paths lead to three different states, each of which matches one of the scenarios. In this example, if T_2 precedes T_1 and T_3 , the cell cycle regulation enters a dead-end state because both DnaA and GcrA levels stay low, leaving no signals to activate CtrA. A regular simulation with set parameter values is equivalent to following one specific path in this tree graph and will never be able to identify a potential failure case such as T_2 . The tool then enters the node T_1 and T_3 separately to compute the next possible states and keep tracking of every single path until it either returns to the root T_0 , or gets stuck in a leaf (dead end) node or a local loop. The later two cases clearly indicate that there are potential hazards in the regulation system and the path leading to the hazard is reported by NuSMV.

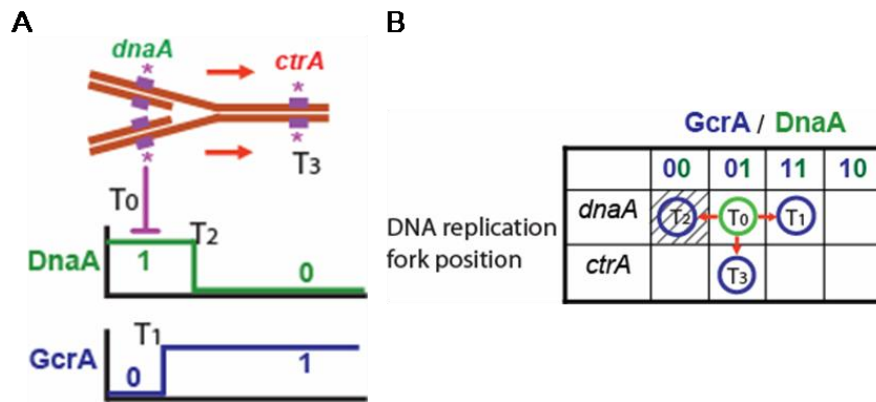


Figure 5.6 A conceptual illustration of how symbolic model checking examines timing. (A) A cartoon depicting the events happening at the four time points: hemi-methylation of the *dnaA* promoter (T_0), GcrA level rise to high (T_1), DnaA level falls to low (T_2), and hemi-methylation of the *ctrA* P1 promoter (T_3). (B) A state transition table showing the three possible paths (red arrows) leading to the successor states from the instant state at T_0 . Each successor state (shown by a circled T_i) corresponds to a next event after T_0 . One of them (shaded T_2) is a undesirable state with no regulators left to continue the cell cycle. The top row indicates the discrete levels of the regulators corresponding to the states and the left column indicates the position of the regulation fork.

The path leading back to T_0 does not guarantee successful completion of the cell cycle. A challenge in any method for assessing biological robustness is defining requirements for successful operation of the modeled system. For the *C. crescentus* cell cycle control, the key functionality of the control circuit is to ensure that actions that affect completion of the cell cycle -- cell growth, chromosome replication, asymmetric cell division, and so forth -- repeatedly occur in the proper order. This includes ordering of expression of the seven proteins in the model (Figure 3.1), switching of the CckA/ChpT phosphosignaling pathway, and activation of DNA replication followed by cell constriction, compartmentalization into nascent swarmer and stalked cell compartments, and cell division. Thus, we required that all feasible orderings of the event T_i 's the model system would successfully produce an unbounded succession of Stalked \rightarrow DNAREplication \rightarrow Compartmentalization \rightarrow (Swarmer OR Stalked) states as occurs in the tree of descendent daughter cells of an initial swarmer cell. This criterion is written in computational tree logic (CTL) to cover all possible paths in the state space {Clarke, 1981 #884; Queille, 1982 #885; Emerson, 1985 #896}. For the subset of cell cycle subsystems we model, and for the asynchronous logic abstraction of the model, satisfaction of this requirement is evidence that the cell cycle control is robust.

A hazard detected by the model checking can be a biological possibility that has acceptably low probability of occurrence, or it may represent a feature of the biological system that prevents the hazard from occurring, but is missing in the simulation model.

5.5 Model checking results

The model checking program exhaustively checked the enormous number of alternative possible Ti combinations, and only two potential hazards were identified. Considering that the discrete regulatory system model has about 400,000 possible discrete states and a far greater number of possible timing scenarios, this result suggested that overall design of the cell cycle control circuit has been optimized by evolutionary selection to operate over a wide range of nutrient conditions and to be resistant to stochastic variations in time to complete various subsystem operations or signaling pathways. Further, it is evidence that conclusions relating to operation of the cell cycle control based on the simulation model do not depend on the exact parameter values for the simulation.

Biological implications of NuSMV identified timing hazards

Hazard scenarios identified by the NuSMV timing analysis could be either (i) actual hazards in the *Caulobacter* cell cycle control that are so unlikely to occur that they do not affect fitness, or (ii) indicative of mechanisms that are in the cell's control system, but missing in the model. Two cases where the timing analysis led to refinements in the model are: The two hazards identified both relate to repressive feedback signals (GcrA repression of DnaB expression and CtrA repression of FtsZ expression, see Figure 3.1). In both of these cases, the hazards are caused by anomalously slow synthesis of the protein coupled with anomalously fast synthesis of the repressive feedback signal, which blocks cell cycle progression. Both cases appear to have relatively low probability of occurring, and there also may be undiscovered aspects of the regulation of both DnaB and FtsZ that eliminate the hazard.

Methylation based control

We also examined the incremental contribution of the methylation-based control of the *ctrA* P1 promoter to robustness of the cell cycle control. When we simulated an *in silico* mutant where

full-methylation does not repress the *ctrA* P1 promoter, the cell cycle operated correctly, however, applying the model checking program to this *in silico* mutant identified an additional hazard: in this mutant, CtrA could be synthesized too early, leading to premature repression of FtsZ and thus failure of cytokinesis. This suggests that although this methylation-based mechanism is not essential, it contributes to fitness of the organism by assuring correct operation of the cell cycle in the fraction of cells where stochastic variation produces anomalously early expression of CtrA.

The DNA methylation-based control of *dnaA* transcription also contributes to robustness of cell cycle control by reducing its expression during DNA replication so that the likelihood of over-initiation of replication is reduced. However, the methylation-based repression of *dnaA* is not complete, so that some expression remains {Collier, 2007 #784}. The remaining expression was initially regarded as an undesirable trait but too insignificant to play any functional role in cell cycle regulation. Moreover, the reported basal expression was suspected to be a measurement artifact having arisen from the loss of synchrony in the cell population and the wild-type simulation generated indistinguishable results with or without the remaining basal expression in the hybrid system model.

The robustness analysis revealed that the remaining basal expression of the methylation-based control plays an important role in cell cycle regulation by avoiding a failure situation analogous to a class of timing hazard called glitch disappearance in electrical circuit design (Figure 5.7A, Figure 5.7B). With nominal kinetic parameter values, there are usually enough GcrA molecules in the cell to activate the *ctrA* P1 promoter when it gets hemi-methylated (Figure 5.7C) during chromosome replication, but unusually slow DNA replication or fast DnaA degradation could give rise to a situation where GcrA is depleted prematurely before hemi-methylation of the *ctrA* P1 promoter (Figure 5.7D). Then, CtrA would not re-accumulate in predivisive cells to activate the synthesis of FtsQA, and cell constriction would not occur.

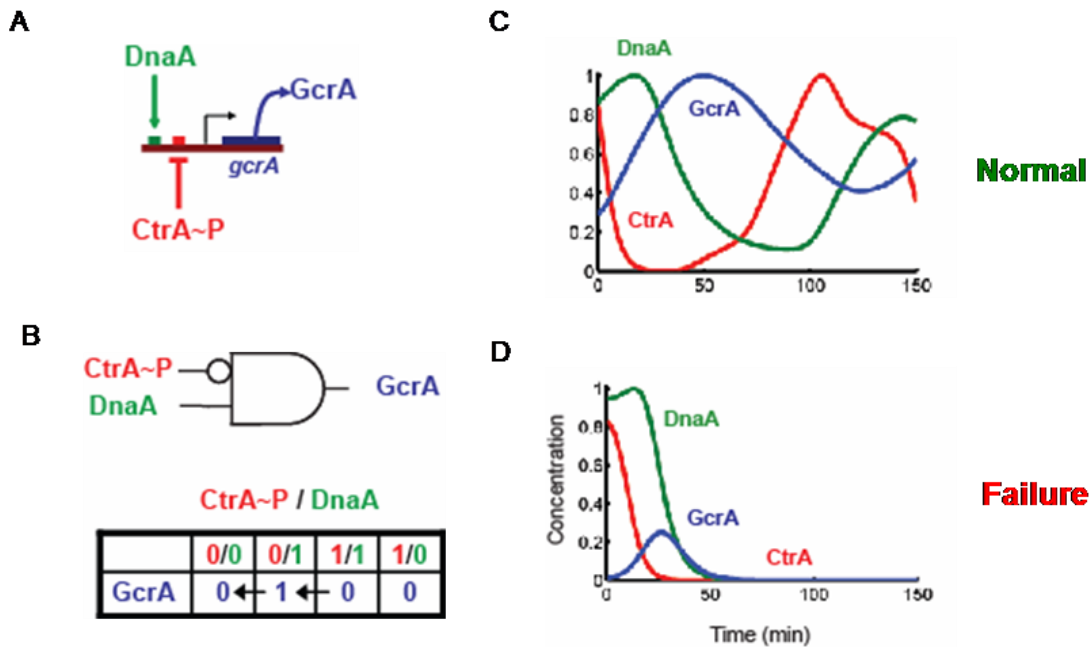


Figure 5.7 A hazard identified by the robustness analysis of a cell cycle model without remaining basal expression of the methylation-based control. (A) The *gcrA* promoter is activated by DnaA and repressed by CtrA~P. (B) A logic gate representation of the promoter. The state transition table shows that GcrA signal only temporarily asserts, forming a pulse or a glitch. (C) A simulation with normal parameter values shows enough GcrA to activate *ctrA* transcription. (D) A simulation with faster DnaA proteolysis confirms the identified hazard by showing that a resulting weaker GcrA pulse causes GcrA to be depleted before activating the hemi-methylated *ctrA* P1 promoter. The hazard has also been confirmed by a simulation with slower DNA replication (not shown).

Symbolic model checking discovered that the low-level synthesis of DnaA by the *dnaA* basal expression, however, can restart GcrA synthesis and, in turn, CtrA synthesis to rescue the cell (Figure 5.8). Mutant strains mimicking the identified failure situation *in vivo* were still able to replicate and divide with a defective phenotype of elongated cells (Figure 5.8A, Figure 5.8B) {Collier, 2007 #784}, because it takes extra time (thus extra growth for the cell before division) for the basal expression to accumulate enough DnaA to activate GcrA and CtrA to rescue the cell cycle from this otherwise deadly hazard (Figure 5.8C, Figure 5.8D). Follow up *in silico* mutant simulation with the remaining basal expression confirmed the experimental observation. So in this case, rather than an undesirable trait or a measurement artifact, the “leaky” expression from a supposedly OFF methylation-based control is actually an evolved design feature to prepare the biological circuit for contingencies.

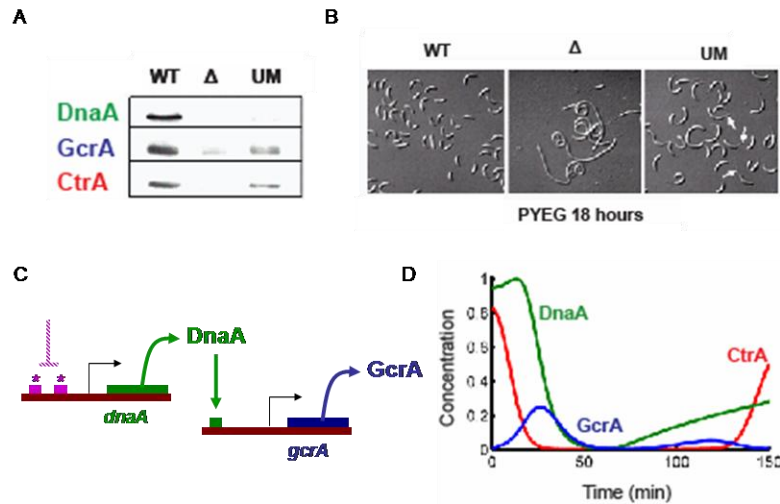


Figure 5.8 Remaining basal expression of the methylation-based control keeps cell cycle regulation robust. (A) Western blots of the three master regulators measured from three *in vivo* strains. The column marked titled WT is the wild-type strain, Δ is a *dnaA* depletion strain, and UM is a strain that keeps the methylation-based control OFF by preventing methylation of the promoter. As a control, the *dnaA* depletion strain (Δ) essentially removes the remaining basal expression and the western blots show that the synthesis of the master regulators is completely stopped. The remaining basal expression of *dnaA* in the UM strain still allows a low level of synthesis as shown by the UM column. (B) Microscopic cell images of the three strains. WT shows the wild-type cells, Δ shows the *dnaA* depletion strain which has stopped dividing and become filamentous, and UM shows elongated but still dividing cells, confirming the western blots in (A) that the remaining basal expression circumvents the hazard. (C) A schematic of the *dnaA* and *gcrA* promoter. Remaining expression from the OFF methylation-based control allows accumulation of DnaA and GcrA. (D) An *in silico* mutant simulation of UM showing how the remaining basal expression of *dnaA* allow accumulation of GcrA and, in turn, CtrA.

Halting and restarting the cell cycle.

The DnaA protein is strategically located in the circuit to simultaneously activate DNA replication and the GcrA-CtrA-CcrM pathway at the beginning of each cell cycle (Figs. 1C, S1) {Hottes, 2005 #782; Gorbatyuk, 2001 #168; Collier, 2006 #783}. Elimination of DnaA halts the cell cycle {Gorbatyuk, 2001 #168}, and control of DnaA stability is a mechanism used to halt the *C. crescentus* cell cycle in response to stress {Gorbatyuk, 2005 #754}. When *C. crescentus* cells are starved for carbon or nitrogen, DnaA is rapidly proteolyzed, and relief of the starvation rapidly restabilizes DnaA {Gorbatyuk, 2005 #754}. We used model checking to assess whether this DnaA stability-dependent method of halting and restarting the cell cycle is robust by adding

an additional switch in the model to modulate DnaA stability. This analysis showed that the cell cycle stops and restarts correctly regardless of when DnaA stability changes during the cell cycle.

In this chapter, formal verification using symbolic model checking has demonstrated the robustness of the *Caulobacter* cell cycle regulation, which leverages various mechanisms to complete cell cycle division under unusual or altering conditions. The hazards identified in the wild type and mutant models provided biological insights into the robustness property of the regulation system, especially the role of methylation-based control of promoters of master regulators.

Furthermore, the roles of the various pathways in the regulatory network can be categorized by examining their corresponding *in silico* depletion mutant using both hybrid simulation and symbolic model checking. If a depletion mutant is unable to complete its cell cycle in the hybrid simulation, its corresponding pathway must be essential to the organism. On the other hand, if no cell cycle defects are observed in the hybrid simulation, but the deletion mutant gives rise to new hazards in symbolic model checking, the corresponding pathway must contribute to the robustness of the cell cycle even though it is not essential. Moreover, if the depletion is neither essential nor contributive to robustness, but results in a lengthier cell cycle, the corresponding pathway must be responsible of enhancing the efficiency of the regulation system, thus improving the fitness of the organism. This exercise helps us understand the structure of the regulatory network from the design perspective. For example, the methylation based promoter controls have been shown to be a robustness mechanism, while some of the regulated proteolysis mainly help shorten the cell cycle.

In addition to helping answer many of the questions about the *Caulobacter* cell cycle, our approach also opened new doors to further exploration. The next chapter will recapitulate the novel aspects of our approach as well as describe potential new research directions.

Chapter 6

Summary and discussion

Caulobacter cell cycle is regulated by a genetic circuit of master regulators which is synchronized to progression of lengthy cell processes through feedback mechanisms such as methylation-based promoter control and phosho-signal pathway. It is still an ongoing process trying to reverse engineer the *Caulobacter* cell cycle regulation system, but a combination of simulation using hybrid system control model and robustness analysis using symbolic model checking presented in this work has greatly enhanced our understanding of the underlying principles of this biological system. Modeling cell cycle control as a hybrid system allowed us to deal with different levels of abstraction to incorporate pathways and cell processes with only incomplete knowledge. A higher level discrete representation of the simulation model was developed for symbolic model checking methods to fully assess the impact of parameter variation on the modeled cell cycle control. With new experimental evidence emerging, constant update and expansion is anticipated for the model to make more detailed predictions.

6.1 Conclusion

Mimicking the *in vivo* control architecture of *Caulobacter* cell cycle regulation, the hybrid system model has been validated by experimental measurements, including western blots, Micorarray, and fluorescent microscopic images, of wild-type and various mutant strains. The simulation confirms that non-transcriptional feedback mechanisms such as methylation-based

control, phospho-signal pathway, and regulated proteolysis are critical for synchronizing the cascade of master regulators to the lengthy cell cycle processes in a “just-in-time” way.

Only included as an abstract binary signal in the simulation model, the phospho-signal pathway is a critical switch for turning on and off CtrA and is an active research area where multiple kinases and phosphatases including CckA, DivK, DivJ, PleC and ChpT have been identified {Jacobs, 1999 #248; Jacobs, 2001 #174; Iniesta, 2006 #785; Quon, 1996 #329; Biondi, 2006 #786}. However, the system level simulation points to two missing pieces of information in the current understanding which are critical for solving the puzzle from the perspective of top level control: (i) which factor triggers the phospho-pathway to unphosphorylate and proteolyze CtrA at swarmer-to-stalked transition; (ii) which factor triggers the pathway to re-localize CckA and phosphorylate CtrA in pre-division cells. The answers to these two questions will have many implications regarding the timing and robustness of the regulation scheme.

Phosphorylation and proteolysis of CtrA originate from the same pathway and were shown by *in silico* and *in vivo* mutants to be redundant processes, without either of which cells are still able to divide. The redundancy, however, provide two advantages for the regulation: robustness and fitness. Perturbation to either one of them does not stop cell cycle progression while combination of the two accelerates the progression by making the CckA-CtrA switch more efficient.

With the exception of CtrA, in the model the regulated proteolysis of the other three master regulators, DnaA, GcrA, and CcrM are currently activated by an abstract signal indicating one of the three cell cycle stages: swarmer, stalked, and pre-division, because they are based on half-live measurements in different cell cycle stages. A more realistic model should include the actual protein regulators that trigger the molecular proteolysis machineries such as ClpXP and Lon to degrade the master regulators. The combination of simulation results with Microarray measurements should provide some clues to these regulators because their expression is supposed to coincide with transition into a new cell cycle stage. Nevertheless, subsequent robustness analysis showed that the proteolysis is not essential to cell cycle because the control is insensitive to their timing. We postulate that they are for fitness reasons -- accelerate cell cycle progression by minimizing unnecessary “wait” time.

A comprehensive robustness analysis using symbolic model checking exhaustively searched the entire state space of the model and showed that the top level control of *Caulobacter* cell cycle regulation is extremely robust. Both identified hazards correspond to highly unlikely situations which might still be compensated by mechanisms *in vivo* not yet included in the model. A careful examination of the regulation system ensures that no properties of interest were lost regarding dynamic operation when an equivalent asynchronous discrete representation was constructed based on the hybrid system model for model checking. Three aspects of cell cycle regulation has enabled this discrete abstraction with timing: (i) The sigmoidal shape of the Hill function approximation of regulator expression levels ensures the output of any cascaded signal pathways always settles into one of the saturation (discrete) regions in the presence of noise. (ii) Over-expression and proteolysis guarantees that regulator levels eventually settles above or below their respective thresholds for activating downstream regulators and the final outcome of the cell cycle regulation is discrete with finite number of states..

Analysis of *in silico* mutants further revealed the important role played by methylation-based control of the promoters. A novel revelation was that the remaining basal expression of methylation-based control serves as a “time out” control for contingency by re-accumulating necessary master regulators when cell cycle inadvertently enters a potential hazard. Robust analysis also verified that *Caulobacter* cell cycle regulation could react to stress and starvation conditions any time during the cell cycle by starting and halting the control circuit completely hazard free. This capability is probably crucial for *Caulobacter* cells to survive in their natural habitat where nutrients are scarce.

6.2 Discussion and future work

The current model only contains ordinary differential equations to model reactions in respect to time. However, polar localization of kinases and proteolysis machineries as well as compartmentalization of the inner membrane has a spatial component to it in addition to time. For instance, there has been hypothesis that the kinase localized at the swarmer pole and the phosphatase localized at the stalked pole work synergistically to form a rapid bi-stable switch, and compartmentalization tips the balance towards opposite directions in the swarmer and the stalked compartments. To model these diffusion processes correctly, partial differential equations

could be used to take concentration gradient within the cell into consideration. Since diffusion is generally much faster than protein synthesis, the more detailed partial differential equations with finer simulation time step are not expected to topple, but rather complement, any conclusions drawn from the current model which treat them as abstract binary signals. These additions could demonstrate the scalability of the hybrid system model.

Asynchronous circuits are usually more challenging to design than synchronous circuits because without a synchronization clock it is very difficult to prevent timing hazards from happening. However, asynchronous circuits only respond when necessary, make them more power efficient, which is an important criteria for biological circuits. To avoid timing hazards, asynchronous designers often follow certain design principles to minimize such risks, and fundamental mode asynchronous state machines (FMAASM) is one of the major design approaches {McClusky, 1986 #809}. In an FMAASM, inputs into a state machine could only changes after the state machine has settled into a stable state. This technique greatly reduces the number of paths in the design space and simplifies the design by avoiding potential hazardous states. A close examination of *Caulobacter* cell cycle regulation suggests that its “design” features resembles an FMAASM because the regulator levels settle and are held steady until a next event triggers one of the feedback mechanisms. So far all evidence has supported this observation, but we still need to understand the missing pieces such as the source for swarmer-to-stalked transition and CckA relocalization before a conclusion can be reached.

So far we have mainly checked robustness in a narrow sense: timing robustness of the control circuit. However, to be robust sometimes also means to be versatile to dramatic environmental changes, where a regulation system has to resort to additional circuitry and switch on/off entire subsystems for alternative strategies. Halting and re-starting cell cycle under stress or starvation is one example. It has been well documented that different modules of genes get turned on when cells live under different kinds of stress conditions or food sources, some may involve small non-coding RNAs {Landt, 2008 #891}. It will be interesting to develop analysis methods to understand how the whole design space is partitioned so that the control circuit works robustly within each partition, but when parameter variation becomes significant enough, the system will switch to a different mode of operation.

As mentioned previously, *Caulobacter* cell cycle regulation is found to be robust and does not depend on the regulated proteolysis of some of the master regulators. On the other hand, mutant strains without these mechanisms will surely lose to wild-type strains in a competition test because the resulting prolonged cell cycle duration. Hence efficiency is also crucial to fitness of a species. Combination of simulation and symbolic model checking has helped us to discern the three layers of cell cycle regulation: essential pathways, mechanisms for robustness, and apparatus for efficiency.

From a design perspective, robustness is often a tradeoff that is associated with a cost. In electrical circuits, additional circuitry means extra area and power consumption. For a biological circuit, what are the extra costs for an additional pathway or mechanism that makes the regulation more robust or more efficient? First, there is energy cost, because it consumes energy to express genes, synthesize and proteolyze proteins. In the case of regulated proteolysis of the master regulators, the cell has to weigh the benefit of the faster growth rate against the higher energy consumption. It makes sense in such circumstances because nutrients are usually available when a cell decides to divide, so the extra energy consumption is a small price to pay for a cell to divide as fast as possible against its competitors before food runs out. Secondly, there is storage cost, because the cell has to spend extra energy and time to copy and maintain additional genomic information for these functionalities. Nevertheless, given the length of a typical genome, it is usually not a significant price to pay for enhancing survivability or competitiveness, as shown by the prevalence of horizontal gene transfer between microorganisms, which provides cells with the building blocks to improve functionalities {Price, 2008 #892}. A counter example may be seen in organisms living in highly stable, predictable or specialized environment where such diversity may not be necessary. Lastly, there is opportunity cost, since these proteins could be rewired to form other functional units, which has been shown to happen frequently during evolution {Price, 2007 #893}. Robustness is only one aspect of a competitive biological organism, which also has to be efficient and versatile to take full advantage of the environment. So evolution pressure indeed behaves like an architect of an engineering system, constantly weighing the various tradeoffs against a customer specification and driving the system to be optimal. A more comprehensive analysis that takes into account all the design features of a biological system would require a further expansion of the system model as well as an enrichment of the current analytical methods.

Appendix A

Regulator Models and Parameters

The online site (<http://www.stanford.edu/group/caulobacter/CellModel>) includes the detailed and commented graphic implementation of the hybrid system model in Simulink and Stateflow. A tutorial is also provided on the same site for readers who want to understand and simulate the model. The tutorial consists of four sections: introduction, description of the continuous regulatory network model, description of the discrete cell function model, and scripts for automated simulation.

A.1 ODE models of master regulators

In the Simulink subsystem approximate kinetic models in the form of ODEs are used to predict the changing intracellular concentration of the regulatory proteins and mRNAs. Table A.1 shows the models within the simulation governing regulation of protein synthesis, proteolysis and activation {Rosenfeld, 2005 #811}. Stochastic effects are neglected in the single cell model as noted above. The level of promoter activation (as a fraction of the maximum activation) is modeled using functions based on a Hill function approach. Protein production (nM/second) is modeled by a multiplicative constant representing the maximum synthesis rate, times the fractional promoter activation. This is equivalent to assuming a constant average rate of protein production per mRNA. The effects of promoter methylation states are included in the promoter activation models, as are the cases where there are multiple promoters or multiple

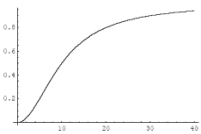
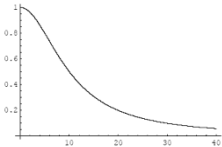
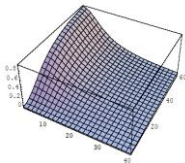
regulatory ligands. Instantaneous values for the binary switch parameters in the ODEs (e.g., the methylation state of a promoter) are determined by the Stateflow subsystem. Proteolysis is modeled by an exponential decay function with a half-life parameter. The timing of initiation of CtrA proteolysis is determined by the Stateflow controlled phosphosignaling pathway originating at CckA. In other cases where there is experimental data for different half-lives at different cell cycle stages, the respective half-life parameters are set to the observed values (by input from the Stateflow subsystem) as the cell cycle progresses.

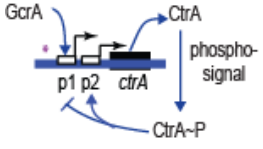
In the model equations in Table A.1, protein and mRNA rates are in nM/sec. Where necessary conversions between molecules/sec per cell and nM/sec per cell were made using


$$M \text{ molecules/sec} = \frac{M}{A_g \cdot V_{avg}} \text{ nM/sec}$$

where A_g is Avagadro's number in nmol^{-1} and V_{avg} is the average value over the cell cycle of the *Caulobacter* cell volume in liters. Since we simulate the molecular concentrations instead of the number of molecules/cell at the time of compartmentalization, concentrations of cytoplasmic proteins are initially equivalent in each of the new compartments.

Table A.1 Models used in the Simulink subsystem for protein production and activation levels

	<p>Activator: $\frac{d[C]}{dt} = \frac{\beta/V}{1 + \left(\frac{[C_d]}{[C_t]}\right)^n}$</p>
	<p>Repressor: $\frac{d[C]}{dt} = \frac{\beta/V}{1 + \left(\frac{[C_t]}{[C_d]}\right)^n}$</p>
	<p>Activator-repressor: $\frac{d[C]}{dt} = \frac{\beta/V}{\left\{1 + \left(\frac{[C_{d1}]}{[C_{t1}]} \right)^{n1}\right\} \left\{1 + \left(\frac{[C_{t2}]}{[C_{d2}]} \right)^{n2}\right\}}$</p> <p>Where n is the Hill coefficient, $[C_d]$ is the concentration of the transcriptional factor that yields half-maximal expression, $[C_t]$ is</p>

	<p>the concentration of the transcriptional factor, β is the maximal protein production rate, V is the volume of a <i>C. crescentus</i> cell. β/V for a specific gene remains approximately constant throughout the cell cycle because the effect of cell growth on concentrations is compensated by the increase of gene copy numbers.</p> <p>In the simulation model, n is a unit-less number, $[C_d]$ and $[C_t]$ are expressed in nM (nanomolar), β/V is expressed in nM/s (nanomolar per second).</p>
	<p>Degradation: $\frac{d[C]}{dt} = -\lambda \cdot [C] = -\frac{\ln 2}{hl} [C]$</p> <p>Where hl is the half-life of the protein in min, and λ is the degradation rate constant in min^{-1}.</p>
<p>Specific forms for individual genes.</p>	
	$\frac{d([CtrA] + [CtrA \sim P])}{dt} = \frac{\beta_1 / V}{\{1 + (\frac{[C_{d1}]}{[GcrA] + \varepsilon})^n\} \{1 + (\frac{[CtrA \sim P]}{[C_{d2}]})^n\}} \cdot [(1 - \eta) \cdot m_{ctrA} + \eta]$ $+ \frac{\beta_2 / V}{1 + (\frac{[C_{d3}]}{[CtrA \sim P] + \varepsilon})^n} - [CtrA](\lambda_1 \cdot clpXP + \lambda_2)$ $\frac{d[CtrA \sim P]}{dt} = (k_f \cdot [CtrA] - k_r [CtrA \sim P])$ <p>We estimate from {Reisenauer, 2002 #104} that expression for the <i>ctrA P1</i> promoter is repressed by about 7-fold when the promoter is fully-methylated compared to when it is hemi-methylated.</p> <p>η is the relative basal expression level of a fully-methylated (OFF) <i>ctrA P1</i> promoter compared to that of a hemi-methylated (ON) <i>ctrA P1</i> promoter. m_{ctrA} is the methylation state of the <i>ctrA</i></p>

	<p>P_1 promoter. $m_{ctrA}=0$ when it is fully-methylated and the maximal $ctrA$ P_1 expression level is $\eta\beta_1$. $m_{ctrA}=1$ when it is hemi-methylated and the maximal $ctrA$ P_1 expression level is β_1. β_2 is the maximal expression level of the $ctrA$ P_2.</p> <p>$clpXP=1$ when the $clpXP$ proteolysis machinery is active and $clpXP=0$ when it is inactive. The m and $clpXP$ binary switches are set by inputs from Stateflow. The infinitesimal value ϵ is added to avoid divide-by-zero errors in the numerical integration routines.</p> <p>Since we assume that the speed of phosphorylation is far greater than the speed of protein synthesis when the phosphosignal is on, we use a linear equation to approximate the Michaelis-Menten rate equation for phosphorylation:</p> $\frac{d[CtrA \sim P]}{dt} = k_1 \cdot cckA \cdot [CtrA] + k_2 \cdot [CtrA] - k_3 [CtrA \sim P]$ <p>k_1 is the phosphorylation rate of CtrA with the CckA phosphosignal present. The $cckA$ binary switch from Stateflow is set to 1 when the CckA phosphosignal is active, and 0 when the CckA phosphosignal is inactive. k_2 is the phosphorylation rate of CtrA without the CckA phosphosignal (presumably very low). k_3 is the de-phosphorylation rate of CtrA~P.</p>
	$\frac{d([GcrA])}{dt} = \frac{\beta/V}{\left\{1 + \left(\frac{[C_{d1}]}{[DnaA] + \epsilon}\right)^n\right\} \left\{1 + \left(\frac{[CtrA \sim P]}{[C_{d2}]}\right)^n\right\}} [GcrA](\lambda_{sw} \cdot isSW + \lambda_2)$ <p>λ_{sw} is decided by the half-life of GcrA in swarmer cells, which is shorter than the half-life of GcrA in the stalked and pre-division cell stages. During the simulation, while the cell is in the swarmer stage, the binary variable $isSW$ from Stateflow is set to 1.</p>



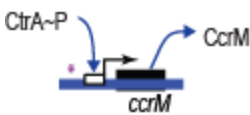
$$\frac{d([DnaA])}{dt} = \frac{\beta / V [(1 - m_{dnaA} + \kappa \cdot m_{dnaA})]}{\left\{1 + \frac{[DnaA]}{[C_{d1}]}\right\} \left\{1 + \left(\frac{[GcrA]}{[C_{d2}]}\right)^n\right\}} - [DnaA](\lambda_{sw} \cdot isSW + \lambda_{st} \cdot isST + \lambda_2)$$

Expression of *dnaA* is repressed by about 3-fold when the *dnaA* promoter is hemi-methylated {Collier, 2007 #784} compared to fully-methylated.

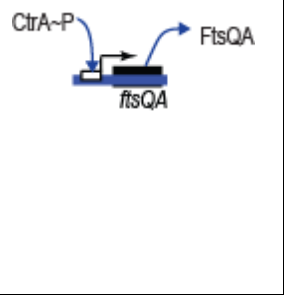

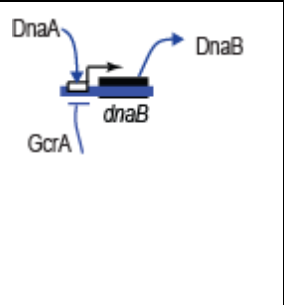
In the simulation model, κ is the relative expression of a hemi-methylated (OFF) *dnaA* promoter compared to that of a fully-methylated (ON) *dnaA* promoter. m_{dnaA} is the methylation state of the *dnaA* promoter. When m_{dnaA} is 0, signifying that *dnaA* is fully-methylated, the maximal expression level is β . If m_{dnaA} is 1, or *dnaA* is hemi-methylated, the maximal expression level is $\kappa\beta$.

Given the that there is one putative DnaA binding site in the sequence upstream of the *dnaA* coding sequence, it has been hypothesized that *dnaA* transcription is auto-regulated by DnaA {Zweiger, 1994 #374}, although this hypothesis does not affect the simulation outcome in any significant way. DnaA synthesis is also repressed by GcrA {Holtzendorff, 2004 #33}, probably by a post-transcriptional mechanism. We have approximated this effect with a Hill function.

DnaA proteolysis is also cell cycle regulated {Gorbatyuk, 2001 #168}. The proteolysis rate is set by the cell cycle stage parameters, *isSW* and *isST*.



$$\frac{d([CcrM])}{dt} = \frac{\beta / V}{\left\{1 + \left(\frac{[C_d]}{[CtrA \sim P] + \varepsilon}\right)^n\right\}} \cdot m_{ccrM} - [CcrM](\lambda_{sw} \cdot isSW + \lambda_2)$$

	$\frac{d([FtsQA])}{dt} = \frac{\beta/V}{\{1 + (\frac{[C_d]}{[CtrA \sim P] + \epsilon})^n\}} - [FtsQA](\lambda_{sw} \cdot isSW + \lambda_2)$ <p>There is a second <i>ftsA</i> only promoter which is relatively weak and ignored by the model {Sackett, 1998 #263}.</p>
	$\frac{d([FtsZ])}{dt} = \frac{\beta/V}{\{1 + (\frac{[C_{d1}]}{[DnaA] + \epsilon})^n\} \{1 + (\frac{[CtrA \sim P]}{[C_{d2}]})^n\}} - [FtsZ](\lambda_{pd} \cdot isPD + \lambda_2)$ <p>The <i>ftsZ</i> promoter is activated by DnaA {Hottes, 2005 #782} and repressed by CtrA~P {Kelly, 1998 #267}.</p> <p>FtsZ has a shorter half-life in the pre-divisional stage which is modeled by an added degradation term activated only in the predivisional cell stage, when <i>isPD</i> is set to 1 by the Stateflow cell stage monitor.</p>
	$\frac{d([DnaB])}{dt} = \frac{\beta/V}{\{1 + (\frac{[C_{d1}]}{[DnaA] + \epsilon})^n\} \{1 + (\frac{[GcrA]}{[C_{d2}]})^n\}} - [DnaB]\lambda$ <p>The <i>dnaB</i> promoter is activated by DnaA {Hottes, 2005 #782} and repressed by GcrA {Holtzendorff, 2004 #33}.</p>

We assume that the effects of growth in the cell volume can be neglected. Cell growth causes continuous dilution of the protein concentration and is thus computationally similar to a protein degradation term with half-life equivalent to the cell generation time (assumed to be 135 min). Since regulatory protein half-lives are observed to be much less than the generation time, this dilution effect is negligible. In the case of protein synthesis, a gene producing a constant average rate of molecules/second would produce twice the incremental protein concentration/sec (nM/sec) in the small initial cell compared to the rate per gene in the larger cell near cell division. However, over the course of the cell cycle, each gene is duplicated so that the gene

dosage is doubled and thus the rate of protein production from activated genes is also doubled compensating for the larger cell volume.

A.2 Parameters

The cell cycle simulation model has a total of 62 parameters. Among them, 29 parameters have experimentally measured values (Table A.2A), 25 parameters have estimated nominal values (Table A.2B), and 9 parameters are used for *in-silico* mutant simulations. The robustness analysis as described in the main text found that the cell cycle control circuit design will execute the cell cycle correctly over wide ranges of parameter values.

We use data from {Keiler, 2003 #91} scaled to a 135 min swarmer cell generation time for the timing of the swarmer cell stage, and chromosome replication. With the exception of CtrA and CcrM, *in vivo* measurements of the number of protein molecules in the *C. crescentus* cell have not been published, and *in vitro* studies to measure ligand-promoter binding kinetics are only available for a few CtrA binding sites {Siam, 2003 #61; Quon, 1998 #274}. Accordingly, we normalize protein concentrations (from quantified Western blots) to the maximum concentration, and we assume that all promoters are activated at a small fraction of the peak level of the activating ligand(s). We used a parameterized Hill-function type model of gene activation that yields good agreement between observed (normalized) mRNA temporal profiles from microarray assays and predicted values from the simulation. Half-lives of DnaA, GcrA, and CtrA have been experimentally determined in swarmer cells and stalked cells {Collier, 2006 #783; Domian, 1997 #290; Gorbatyuk, 2005 #754}, and the active regulation of CtrA proteolysis as a function of the cell cycle has been extensively studied {McGrath, 2006 #859; Iniesta, 2006 #785}. The relatively small dilution effects of cell growth are assumed to be included in the experimental protein half-life data. Pathways controlling DnaA and GcrA stability have not been characterized, so we modeled the observed dynamic control of DnaA and GcrA stability by setting their half-lives to the reported value at each stage in the cell cycle. Time-resolved measurements of protein phosphorylation states are possible, but with poor resolution, and *in vivo* kinetics of *C. crescentus* phosphorylation reactions are not available. We assume that the phosphosignaling reactions are fast enough that phosphorylation-related switching is much faster

than switching by genetic mechanisms or protein degradation. The rationale for choice of all parameter values is in Section C.

Naming conventions for Hill function parameters are as follows:

The default value of Hill coefficient n for all promoters is denoted by $hcDef$ (In the current model, all the promoters share the same default Hill coefficient value). The default concentration of a transcriptional factor ($[C_d]$) that yields half-maximal expression is denoted by $cHalfDef$. The concentration of CtrA~P that yields half-maximal expression in the CtrA~P-regulated genes is different from the default value $cHalfDef$ and is denoted by $cHalfCtrA$. The maximum protein synthesis rate divided by the cell volume, β/V , is converted to nanomolar per second (nM/s) and denoted by pX for protein X. The half-life of protein X is denoted by hlX . The initial concentration of protein X is denoted by $cX0$.

For comparison to experimental values, estimated protein concentration profiles are normalized to a peak value of one, and experimental values are normalized the same way. Ligand activation levels at downstream binding sites are assumed to be a small fraction of the peak concentration of that ligand.

Concentration vs. number of protein molecules per cell

In the model, we chose to simulate the concentration levels rather than the number of protein molecules per cell. The number of protein molecules per cell is converted to the protein concentration, using:

$$[C] = \frac{M}{A_g \cdot V_{avg}} \quad (A.1)$$

where $[C]$ is the protein concentration in nM (nmol/L), M is protein molecules per cell, A_g is the Avogadro's number ($6.022 \times 10^{14} \text{ nmol}^{-1}$), and V_{avg} is the average volume over a cell cycle of a *C. crescentus* cell in liters.

During cell cycle progression, the *C. crescentus* cell grows larger. In the simulation, we used the average volume of the *C. crescentus* cell to calculate the molecular concentrations. The doubling of the cell volume over the cell cycle is compensated by the doubling of gene copy

number as well by DNA replication. The dilution of protein concentration by growth is comparable to a 135 min half life (for our assumed swarmer cell cycle time) which is well longer than the regulatory protein half lives and thus can be neglected.

During cell cycle progression, the volume of a *C. crescentus* cell grows from around $0.5 \mu\text{m}^3$ in early swarmer stage to around $1.2 \mu\text{m}^3$ in pre-divisional stage. We take $0.75 \mu\text{m}^3$ as V_{avg} .

$$\text{Therefore, } [C] = \frac{M}{A_g \cdot V_{\text{avg}}} = \frac{M}{6.022 \times 10^{14} \cdot 7.5 \times 10^{-16}} = 2.2M \text{ (nM)} \quad (\text{A.2})$$

Notes for Table A.2B -- parameters with estimated values

Measurements of these parameters are currently unavailable so we estimated nominal values. The simulation predictions are not sensitive to most of these parameters for reasons outlined below.

These parameters fall into six categories.

1. Initial regulatory protein concentrations

The cell cycle simulation is relatively insensitive to these initial levels, because after one or two simulated cell cycles the concentrations in swarmer cells stabilize with other values which are determined by other parameters in the model.

2. Protein concentration ranges of action (or thresholds) that control cell functions such as DNA replication, cytokinesis and DNA methylation.

Many cell functions are initiated when a regulatory protein is synthesized and its level rises about a range of action at a downstream binding site. We selected the thresholds to be well below the peak levels (assumed to be the usual situation for bacterial genetic regulatory links). Change in an assumed threshold value will change the time of a regulatory reaction. The sensitivity of cell cycle outcome to timing variations was explored and shown to be low as part of the robustness analysis as described in the main text.

3. Protein synthesis rates from an activated gene

The protein synthesis rate depends on both the rate of transcript initiation and the average number of proteins produced from each mRNA. Both these rates vary in both average values for different genes and stochastically between cells in a population at any given instant. We used nominal synthesis rates chosen to be representative for bacterial promoters. Differences in actual values would affect delays in downstream gene activation or repression and the peak values of protein concentrations in the simulation. Since comparisons to experimental values are done by comparing the pattern of normalized predicted and experimental values, the peak concentration values are not significant. Doubling of gene dosage and offsetting dilution by cell growth are also factors to be considered. Our confidence in this approach was reinforced by the agreement between the simulation predictions of patterns of mRNA and protein concentrations with experimental values (Fig. 2 and <http://www.stanford.edu/group/caulobacter/CellModel>). Further, the robustness analysis showed that the cell cycle circuit design has evolved by selection to be insensitive to variations in signal pathway timing.

4. Hill function parameters

We used the same concentration for half-maximal expression for all proteins except for CtrA as a transcriptional factor, for which we used a slightly higher concentration for half-maximal expression, based on the measured concentration of CtrA in the *C. crescentus* cell. We used a Hill coefficient equal to 2 for all protein regulators.

5. Protein half-lives

The half-life of DnaB has not been experimentally measured. Furthermore, DnaB represents a collection of replication initiation proteins. As long as this half-life value is not extremely small, it has no effect on the simulation outcome. In other words, as long as DnaB or the other replication initiation proteins it represents are not degraded too rapidly during the initiation of chromosome replication (~ 5 minutes), the model is not sensitive to this value.

6. Rate constants for phosphorylation of CtrA

Phosphorylation is assumed to be fast relative to protein synthesis and degradation and thus to act as a rapid switch, so within this constraint the simulation is not sensitive to the specific phosphorylation rate parameters.

With the exception of CtrA and CcrM, *in vivo* measurements of the number of protein molecules in the *C. crescentus* cell have not been published, and *in vitro* studies to measure ligand-promoter binding kinetics are only available for a few CtrA binding sites {Siam, 2003 #61; Quon, 1998 #274}. Accordingly, we normalize protein concentrations (from quantified Western blots) to the maximum concentration, and we assume that all promoters are activated at a small fraction of the peak level of the activating ligand(s). We used a parameterized Hill-function type model of gene activation that yields good agreement between observed (normalized) mRNA temporal profiles from microarray assays and predicted values from the simulation (<http://www.stanford.edu/group/caulobacter/CellModel>).

Table A2: Parameter values

Table A.2.A Model parameters with experimentally measured values

Symbol	Parameter	Value	Units	Source or rationale
cCcrM0	Concentration of CcrM in swarmer cells	0	nM	{Stephens, 1996 #326}
cCtrA0	Concentration of CtrA in swarmer cells	20900	nM	9500 CtrA molecules are present in the swarmer cell {Quon, 1996 #329} or about 20900 nM.
cFtsA0	Concentration of FtsA in swarmer cells	0	nM	{Martin, 2004 #17}
cFtsZ0	Concentration of FtsZ in swarmer cells	0	nM	{Kelly, 1998 #267}
cGcrA0	Concentration of GcrA in swarmer cells	0	nM	{Holtzendorff, 2004 #33}
cMethylCcrM	Concentration of CcrM during chromosome methylation	6600	nM	3000 molecules in the late PD cell {Berdis, 1998 #268} or about 6600 nM.
hlCcrMc	CcrM half-life	15	min	{Wright, 1996 #318}
hlCtrAc	CtrA half-life	52	min	Measured in mixed population {McGrath, 2006 #859}.
hlCtrAf	CtrA half-life during proteolysis by ClpXP	3	min	CtrA half-life is less than 5 minutes in the stalked cell {Domian, 1997 #290}
hlDnaA_sw	DnaA half-life in the swarmer cell	45	min	{Gorbatyuk, 2005 #754}
hlDnaAc	DnaA half-life in the stalked cell	100	min	{Gorbatyuk, 2005 #754}
hlDnaAc_starve	DnaA half-life during starvation	10	min	The half-life of DnaA is 10 minutes during carbon starvation and 15 minutes during nitrogen starvation {Gorbatyuk, 2005 #754}.
hlFtsAc_st	FtsA half-life in the stalked cell	55	min	{Martin, 2004 #17}
hlFtsAc_sw	FtsA half-life in the swarmer	13	min	{Martin, 2004 #17}

	cell			
hlFtsZc	FtsZ half-life in the swarmer cell and the stalked cell	80	min	{Kelly, 1998 #267}
hlFtsZc_pd	FtsZ half-life in the pre-divisional	20	min	{Kelly, 1998 #267}
hlGcrAc_st	GcrA half-life in the stalked cell	42	min	{Collier, 2006 #783}
hlGcrAc_sw	GcrA half-life in the swarmer cell	10.5	min	{Collier, 2006 #783}
pctrAP1MethRatio	The ratio of transcription rates between the fully-methylated and the hemi-methylated <i>ctrA</i> P1 promoter.	0.15		{Reisenauer, 2002 #104}
pdnaAMethRatio	The ratio between the hemi-methylated transcription rate and the fully-methylated transcription rate from the <i>dnaA</i> promoter.	0.3		{Collier, 2007 #784} This ratio is used in the model to approximate the transcription rate of the hemi-methylated <i>dnaA</i> promoter.
tcckA_reloc	Time between the initiation of chromosome replication and CckA re-localization	60	min	{Iniesta, 2006 #785} CckA is relocalized approximately 80 minutes into the cell cycle. The swarmer stage takes 20 so CckA relocalizes 60 minutes into chromosome replication.
tchro	Time required for chromosome replication	80	min	{Keiler, 2003 #91} Scaled for a 135 minute cell cycle.
tftzRing	Time required for cytokinesis	30	min	{Thanbichler, 2006 #894}
tftzRingPinOff	Time between the start of constriction and compartmentalization	12	min	{Judd, 2003 #69; Thanbichler, 2006 #894}
tMethylWindow	Average time for the DNA to be methylated once CcrM concentration reaches cMethylCcrM	10	min	Since there is a 20 minute window {Berdis, 1998 #268} when DNA is being methylated by a high concentration of active CcrM molecules, we chose 10 minutes as the average time it takes for the chromosome to be methylated.
tsw2st	Duration of the swarmer stage	20	min	{Keiler, 2003 #91} Scaled for a 135 minute cell cycle.
zpctrA	The relative location of the <i>ctrA</i> gene on the chromosome	0.3		0 is the <i>ori</i> , and 1 is the <i>terminus</i> of the chromosome {Nierman, 2001 #177}.
zpccrM	The relative location of the <i>ccrM</i> gene on the chromosome	0.25		0 is the <i>ori</i> , and 1 is the <i>terminus</i> of the chromosome {Nierman, 2001 #177}.
zpdnaA	The relative location of the <i>dnaA</i> gene on the chromosome	0.1		0 is the <i>ori</i> , and 1 is the <i>terminus</i> of the chromosome {Nierman, 2001 #177}.

Table A.2.B Model parameters with estimated values

Symbol	Parameter	Value	Units	Source or rationale
--------	-----------	-------	-------	---------------------

cChroCtrA	CtrA threshold level for initiating DNA replication	110	nM	There are 5 CtrA binding sites in the <i>Cori</i> . CtrA blocks DNA replication by binding to these sites {Quon, 1998 #274}. 50 CtrA molecules per cell are assumed to be adequate to ensure blocking, which translates into 110 nM.
cChroDnaA	DnaA threshold level for initiating DNA replication	440	nM	DnaA is necessary for replisome assembly {Gorbatyuk, 2001 #168}. The simulation is not sensitive to this value as long as it is a small fraction of the maximum DnaA level. 440 nM corresponds to 200 molecules per cell.
cChroDnaB	DnaB threshold for initiating DNA replication	440	nM	This concentration is equivalent to 200 molecules per cell.
cCtrAP0	Initial concentration of CtrA~P	10450	nM	This concentration is consistent with the observation that phosphorylation of CtrA is less active in the swarmer stage {Jacobs, 2003 #84}.
cCytoFtsQA	FtsQA threshold for initiating cytokinesis	660	nM	This concentration is equivalent to 300 molecules per cell.
cDnaA0	Initial concentration of DnaA	2200	nM	This concentration is equivalent to 1000 molecules per cell.
cDnaB0	Initial concentration of DnaB	2200	nM	This concentration is equivalent to 1000 molecules per cell.
cHalfCtrA	Concentration of CtrA~P that yields half-maximal expression of CtrA-regulated genes.	1760	nM	This concentration is equivalent to 800 molecules per cell.
cHalfDef	Default level of transcriptional factors that yields half-maximal expression of target genes	660	nM	This concentration is equivalent to 300 molecules per cell.
cZringFtsZ	Minimum level of FtsZ required for forming the FtsZ ring	660	nM	This concentration is equivalent to 300 molecules per cell.
hcDef	Default Hill function coefficient	2		This default value is assumed for all protein synthesis in the model, including CtrA. For future model development, promoter-specific hill coefficient can be specified as hcX for protein X.
hlCcrMc_sw	CcrM half-life in the swarmer cell	3	min	The half-life of CcrM in Table 2A was measured in a mixed population. Western blot of CcrM shows that CcrM is fully depleted in the swarmer cell in a relatively short time, so the half-life of CcrM is estimated to be 3 minutes in the swarmer cell.
hlDnaAc_degrade	DnaA half-life during chromosome replication initiation	15	min	The levels of DnaA decrease during the early stage of chromosome replication, as seen on Western blots of DnaA as a function of the cell cycle.
hlDnaBc	DnaB half-life	10	min	DnaB in this model represents a collection of proteins that are responsible for initiating chromosome replication.

hlPhos_f_e	Time for half of the CtrA molecules to be phosphorylated when the <i>cckA</i> signal is ON.	0.1	min	The value is chosen because phosphorylation reactions are much faster than transcription and degradation reactions.
hlPhos_f_ne	Time for half of the CtrA molecules to be phosphorylated when the <i>cckA</i> signal is OFF.	150	min	CtrA is not actively phosphorylated without the active kinase CckA. {Jacobs, 1999 #248}
hlPhos_r_np	Time for half of the CtrA~P molecules to become dephosphorylated without phosphatase present	3	min	We assume that CtrA gradually becomes inactive when the <i>cckA</i> signal is OFF.
p1ctrA	Maximum synthesis rate of CtrA from the <i>ctrA</i> P1 promoter	6.6	nM/s	The maximum expression level of p2ctrA is roughly 3 times that of p1ctrA {Reisenauer, 2002 #104}.
P2ctrA	Maximum synthesis rate of CtrA from the <i>ctrA</i> P2 promoter	19.8	nM/s	The maximum expression level of p2ctrA is roughly 3 times that of p1ctrA {Reisenauer, 2002 #104}.
pccrM	Maximum synthesis rate of CcrM from the <i>ccrM</i> promoter	13.2	nM/s	Estimated from western blots and {Berdis, 1998 #268}
pdnaA	Maximum synthesis rate of DnaA from the <i>dnaA</i> promoter	1.76	nM/s	The absolute value does not affect simulation outcome after normalization.
pdnaB	Maximum synthesis rate of DnaB from the <i>dnaB</i> promoter	1.1	nM/s	The absolute value does not affect simulation outcome after normalization.
pftsA	Maximum synthesis rate of FtsA from the <i>ftsQA</i> promoter	1.1	nM/s	The absolute value does not affect simulation outcome after normalization.
pftsZ	Maximum synthesis rate of FtsZ from the <i>ftsZ</i> promoter	1.1	nM/s	The absolute value does not affect simulation outcome after normalization.
pgcrA	Maximum synthesis rate of GcrA from the <i>gcrA</i> promoter	2.2	nM/s	The absolute value does not affect simulation outcome after normalization.

Appendix B

Simulated mRNA levels vs Microarray measurements

We used a parameterized Hill-function type model of gene activation that yields good agreement between observed (normalized) mRNA temporal profiles from microarray assays and predicted values from the simulation.

mRNA predictions versus observed

This appendix provides a comparison of simulated mRNA levels versus time in the *Caulobacter crescentus* cell cycle and experimentally observed mRNA levels. All mRNA levels are normalized to a maximum value of 1. With the exception of the *ctrA* P₁ and P₂ promoters, mRNA levels for comparison are from microarray gene expression assays performed on periodic samples from synchronized *Caulobacter* cell {McGrath, 2007 #876}. We the mRNA levels from the microarray compare data to the promoter activation levels modeled by Hill functions in the simulation model.

The ability to predict the distinctive molecular level progression of the cell cycle into the swarmer and stalk daughter cell compartments is a unique aspect of the simulation model. Experimental observations from synchronized populations (e.g., from microarray assays) are averages over many cells, and measurements later in the synchronized cell cycle always include signals from both swarmer and stalked daughter cells. Our single cell predictions of protein and

mRNA levels are made comparable to observations in synchronized cell populations by (i) averaging the predictions from the swarmer and stalked daughter cell branches of the simulation, and (ii) convolving the result with a Gaussian distribution with a 5 minute standard deviation.

The Affymetrix microarray data for mRNA levels and the simulation predictions for mRNA levels are presented in the layout shown below in Figure B.1.

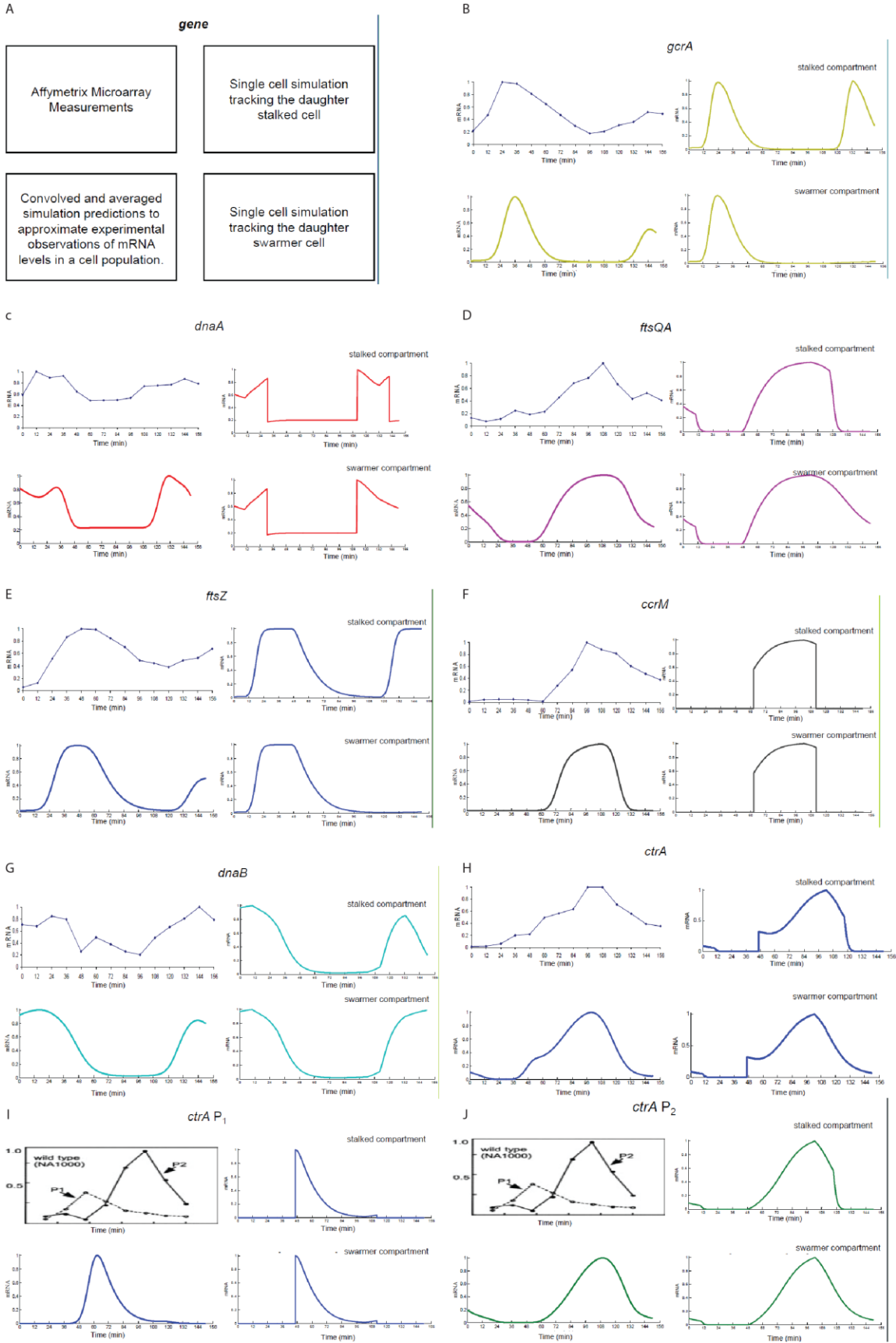


Figure B.1 Simulated mRNA levels vs. Microarray measurements (A) The layout for presenting the Affymetrix microarray data for mRNA levels and the simulation predictions for mRNA levels. (B) Comparison for *gcrA* expression (C) Comparison for *dnaA* expression (D) Comparison for *ftsQA* expression (E) Comparison for *ftsZ* expression (F) Comparison for *ccrM* expression (G) Comparison for total *ctrA* expression. The total *ctrA* mRNA level combines both the *ctrA P1* and *ctrA P2* expression. (I) Comparison for *ctrA* P1 activation level. The top-left plot shows the expression levels of *ctrA* P1 and *ctrA* P2 separately during the cell cycle using a pulse-label immunoprecipitation assay of Beta-galactosidase {Reisenauer, 2002 #104}. (J) Comparison for *ctrA* P2 activation level.

In the simulation model we assume that *ccrM* expression is completely switched off when enough CcrM molecules are synthesized and the chromosomes are remethylated. The actual mechanism for terminating *ccrM* expression is somewhat puzzling in that it is repressed when remethylated, and it would seem that this could happen immediately upon first production of CcrM. The higher levels of mRNA in the late cell seen in the microarray data probably is the result of loss of synchronization of the cell population in the late portion of the experiment.

Appendix C

In silico mutant simulation

This section describes *in silico* mutant strain simulations that emulate several laboratory *C. crescentus* mutant strains. The objectives of these comparisons were to validate the simulation model and to understand the reasons for the experimentally observed phenotypes in greater depth. Additional details of these simulations including graphs of the *in silico* simulation datasets are online at <http://www.stanford.edu/group/caulobacter/CellModel>.

Table C.1 shows the four mutants that were simulated and the changes that were made to the wild-type model to create the mutant simulation. In each case, the simulation predicts (i) the concentration profile of each protein in the model in single cells as a function of cell cycle time when followed into either the swarmer or the stalked compartment of the predivisional cell, (ii) whether the cell can progress through each stage of the cell cycle, and (iii) whether DNA replication and cytokinesis occur normally.

The simulation predictions for all cases checked were consistent with the *in vivo* phenotypes. Simulation results and their relation to the mutant strains in Table S4 are as follows: *GcrA depletion strain*: The simulation predicts that CtrA~P will not re-accumulate after the stalked cell stage, so FtsQA does not accumulate enough to initiate cytokinesis. As a result, the cell cycle arrests at the stalked cell stage in the simulation. The simulated levels of DnaA and CtrA suggest that DNA replication may still happen in these cells before cell death. *Strain with constitutive accumulation of CcrM*: The simulation predicts that the re-accumulation of CtrA in

pre-divisional cells will be delayed ~20 minutes, while DnaA will accumulate at high concentrations throughout the cell cycle. This suggests that over-initiation of DNA replication may take place in these mutant cells as is observed experimentally. As a consequence, the re-accumulation of FtsQA and cytokinesis will be delayed. Strain with *constitutive accumulation of CtrA~P*: The simulation predicts that accumulating CtrA~P will block the initiation of DNA replication. Since cytokinesis is blocked when DNA replication is blocked, the cells will arrest after the stalked cell stage as observed *in vivo*. Strain with the *ctrA* gene moved next to the DNA replication terminus: The simulation predicts that CtrA re-accumulation in predivisional cells will be delayed by ~15 minutes as is observed. As a consequence, the synthesis of FtsQA and cytokinesis will also be delayed, so that the cell cycle will be slightly longer than for wild-type cells. The consistency between the predictions from simulation of the *in silico* mutants and the *in vivo* phenotypes (Table S4) provides additional evidence that our model corresponds to the biological cell cycle control circuitry. Moreover, the predictions from *in silico* mutant simulations provide quantitative insights into how the cell cycle is affected by a given mutation.

The simulations are performed using the Matlab-based simulation of the wild-type *C. crescentus* cell cycle control system. We use the same differential equations, parameter values, and initial conditions as for wild-type cells, except for those parameters that are changed to simulate a mutation of interest. Time varying intracellular concentration levels are predicted for the eight different proteins included in the model.

Table C.1 mutant phenotypes

Characteristics of mutant strains	Genotypes of mutant strains	Refs	Phenotypes <i>in vivo</i>	Parameters changed from wild-type parameters for mutant simulations
Strain where GcrA can be depleted (LS3707)	CB15N $\Delta gcrA$ $P_{xyl}::gcrA$	{Holt zendo rff, 2004 #33}	The cell cycle is arrested at the stalked cell stage, and cells finally die in the absence of GcrA.	Maximum GcrA synthesis rate from the <i>gcrA</i> promoter $pgcrA=0nM/s$
Strain that accumulates CcrM constitutively (LS1)	CB15N $PlacZ::ccrM$	{Zwe iger, 1994 #373}	Cells are slightly elongated and accumulate supplementary copies of the chromosome.	CcrM synthesis rate from a constitutive promoter added in the model $pccrMoe=100nM/s$
Strain that can accumulate stable and constitutively active mutant CtrA proteins	CB15N $pXylX::ctrAD51E\Delta3\Omega$	{Dom ian, 1997 #290}	Cells do not initiate DNA replication and do not divide. Cells elongate before dying.	Binary switch controlling the phosphorylation state of CtrA and the protein half-life of CtrA under active proteolysis by ClpX $isAlwaysCtrAP=1$ $hlCtrAf=200\text{ min}$
Strain where the <i>ctrA</i> gene is moved to a position next to the terminus of replication of the chromosome (LS3355)	CB15N $ctrA\Delta2::pAR358$	{Reis enauer, 2002 #104}	Cell size is sometimes irregular	Relative location of the <i>ctrA</i> gene on the chromosome $zpctrA=1$

Figure C.1 shows the simulation results from the hybrid system model for a wild-type strain, which serves as a reference for the discussion of mutant simulations.

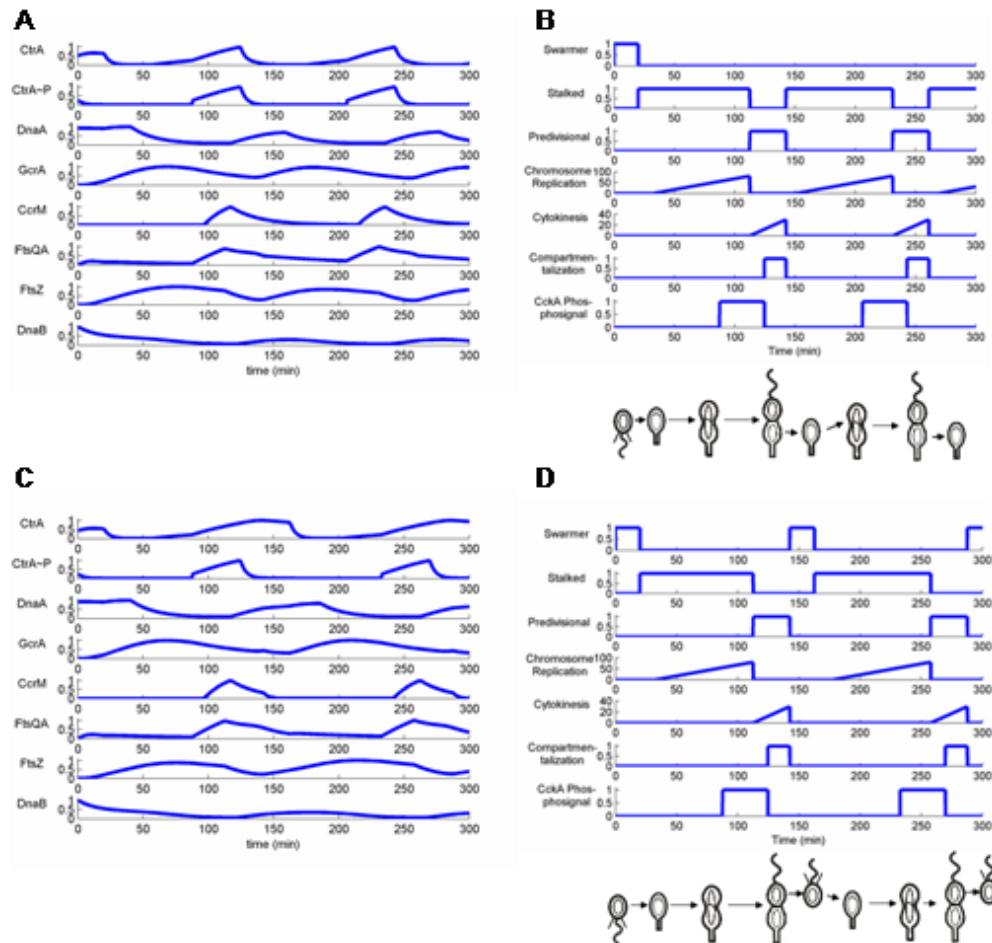


Figure C.1 Wild-type simulation (A) Protein concentration tracking into the stalked compartment. (B) Outputs from Stateflow tracking into the stalked compartment. (C) Protein concentration tracking into the swarmer compartment. (D) Outputs from Stateflow tracking into the swarmer compartment.

1. GcrA depletion strain

C. crescentus mutant previously constructed

A mutant strain expressing the *gcrA* gene conditionally was previously constructed {Holtendorff, 2004 #33}. In this strain ($\Delta gcrA$ *P_{xyI}::gcrA*), the *gcrA* gene was deleted from its native location on the chromosome, but the strain carries a copy of the *gcrA* gene under the control of the xylose-inducible *P_{xyI}* promoter. When a this mutant strain is grown in a media

containing glucose, which turns off the *P_{xyl}* promoter, all cells in the population are arrested as stalked cells, and finally die about 6 hours after the switch to glucose media. It was also observed that CtrA levels become very limiting before cell death, while DnaA levels increase, when GcrA is depleted.

Changes of model parameters to simulate this mutant

To simulate this mutant *in silico*, we used the same equations (Table A.1), the same initial protein concentrations, and the same parameters (Table A.2A and A.2B) as for wild-type cells, except that we changed the value of the “pgcrA” parameter (maximum GcrA synthesis rate from the *gcrA* promoter) in Table A.3C from 6.9 nM/s to 0 nM/s at time 0 min.

Results of the mutant simulation

In this mutant simulation, we observed that the levels of GcrA remain null at all times of the cell cycle, which prevents the activation of the *ctrAP1* promoter by GcrA in stalked cells, and therefore the re-accumulation of CtrA after the stalked cell stage of the cell cycle. As a consequence, CtrA is not present in cells after the stalked cell stage so it cannot activate the synthesis of CcrM and FtsQA (Fig. C.2). The absence of FtsQA after the stalked cell stage blocks progression of cell constriction, so the cells cannot become pre-divisional cells. All these results are in agreement with the phenotype observed *in vivo* {Holtzendorff, 2004 #33} and the simulation helps explain, at a molecular level, how the cell cycle of a single cell is arrested at the stalked cell stage *in vivo* when GcrA is depleted.

We also observed that the minimal levels of DnaA and DnaB during the mutant cell cycle are not as low as the wild-type cells, while the CtrA~P level is insignificant after 40 minutes into the simulation. Since the levels of DnaA and DnaB are over the minimal values required for DNA replication (“cChroDnaA” and “cChroDnaB” are above 1380nM), we predict that it could lead to over-initiation of DNA replication events in the arrested stalked cells. The model does not simulate such events, because of the assumption that no event of initiation of DNA replication could take place if CtrA cannot re-accumulate in cells after the stalked cell stage.

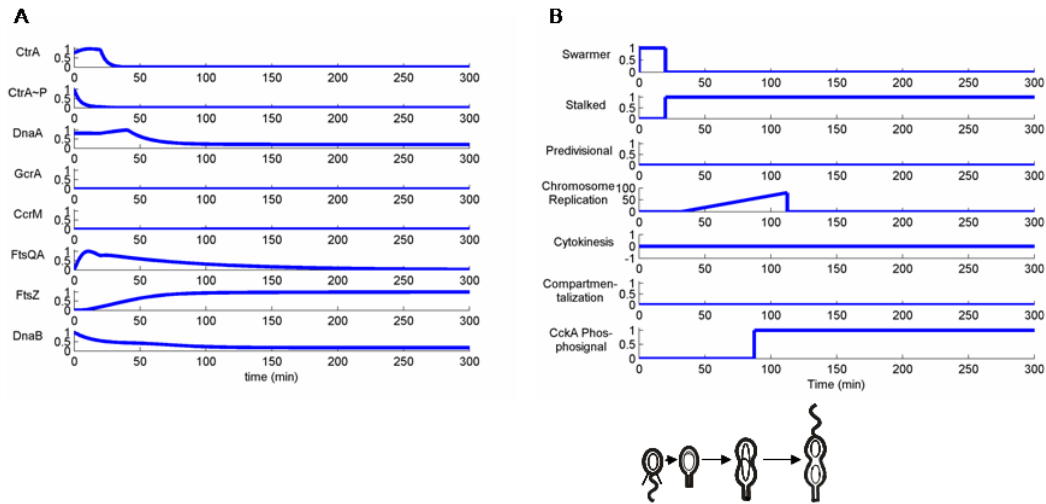


Figure C.2 *In silico* mutant simulation of the GcrA depletion strain. (A) Protein concentration timelines (B) Outputs from Stateflow.

2. *ccrM* constitutively expressed

C. crescentus mutant previously constructed

A mutant strain that expresses *ccrM* constitutively was previously constructed {Zweiger, 1994 #373}. This strain contains a second copy of the *ccrM* gene under the control of the constitutive *PlacZ* promoter integrated at the *ccrM* locus on the chromosome. These mutant cells are sometimes longer than wild-type cells, they sometimes constrict asymmetrically, and their cell cycle is slightly slowed down. These cells also often accumulate more than two chromosomes, showing a defect in the control of chromosome replication initiation.

Changes of model parameters to simulate this mutant

To simulate this mutant *in silico*, we used the same equations (Table S2), the same initial protein concentrations and the same parameters (Table S3A and S3B) as for wild-type cells, except that we added a constitutively activated promoter by setting the “pccrMoe” parameter (CcrM synthesis rate from the constitutive promoter) in Table S3C to a value greater than 0. In the simulation shown here, “pccrMoe” was set to 100nM/s at time 0 min.

Simulation result: Tracking the stalked compartment:

In this mutant simulation, we observed that the levels of the DNA methylase CcrM remain high at all times of the cell cycle, which maintains the *dnaA* and the *ctrAP1* promoters fully-methylated at all times during the simulation. As a consequence, the re-accumulation of CtrA in pre-divisional cells is delayed ~20 minutes, while the DnaA accumulates at high concentrations throughout the cell cycle.

The delayed re-accumulation of CtrA~P in pre-divisional cells in turn delays the accumulation of FtsQA, which delays progression of cell constriction during the mutant cell cycle, compared to the wild-type cell cycle. Since progression of cell constriction is delayed, the overall mutant cell cycle is ~20 minutes longer than the wild-type cell cycle in our simulations, which is in agreement with the phenotypes observed *in vivo* {Zweiger, 1994 #373}. Although the simulation model does not simulate cell growth, we predict that mutant cells will be elongated since their cell cycle is slower but their growth should not be affected compared to wild-type cells. This logical prediction is also in agreement with the phenotypes observed *in vivo*.

The high accumulation of DnaA in the mutant simulation promotes the accumulation of DnaB at a concentration which is above the minimal threshold that is necessary for the initiation of DNA replication (“cChroDnaB” is 1380 nM). The high accumulation of DnaA and DnaB, together with the delayed accumulation of CtrA~P, could promote additional initiation of DNA replication events in these mutant cells. This quantitative simulation would explain why multiple chromosomes are observed to accumulate in mutant cells *in vivo* {Zweiger, 1994 #373}.

Simulation result: Tracking the swarmer compartment:

In this mutant simulation, we observe that the levels of CcrM remain high at all times of the cell cycle, except in swarmer cells (f. Since DNA replication is still efficiently repressed by CtrA~P in swarmer cells (“cChroCtrA” is 345nM), the chromosome is never hemi-methylated by the passage of the replication fork in swarmer cells, even if the CcrM DNA methylase does not accumulate efficiently in swarmer cells. Like during the stalked compartment simulation, the *dnaA* and the *ctrAP1* promoters are maintained fully-methylated at all times of the cell cycle, so the results of the phenotype simulations are comparable when following both compartments of the cell.

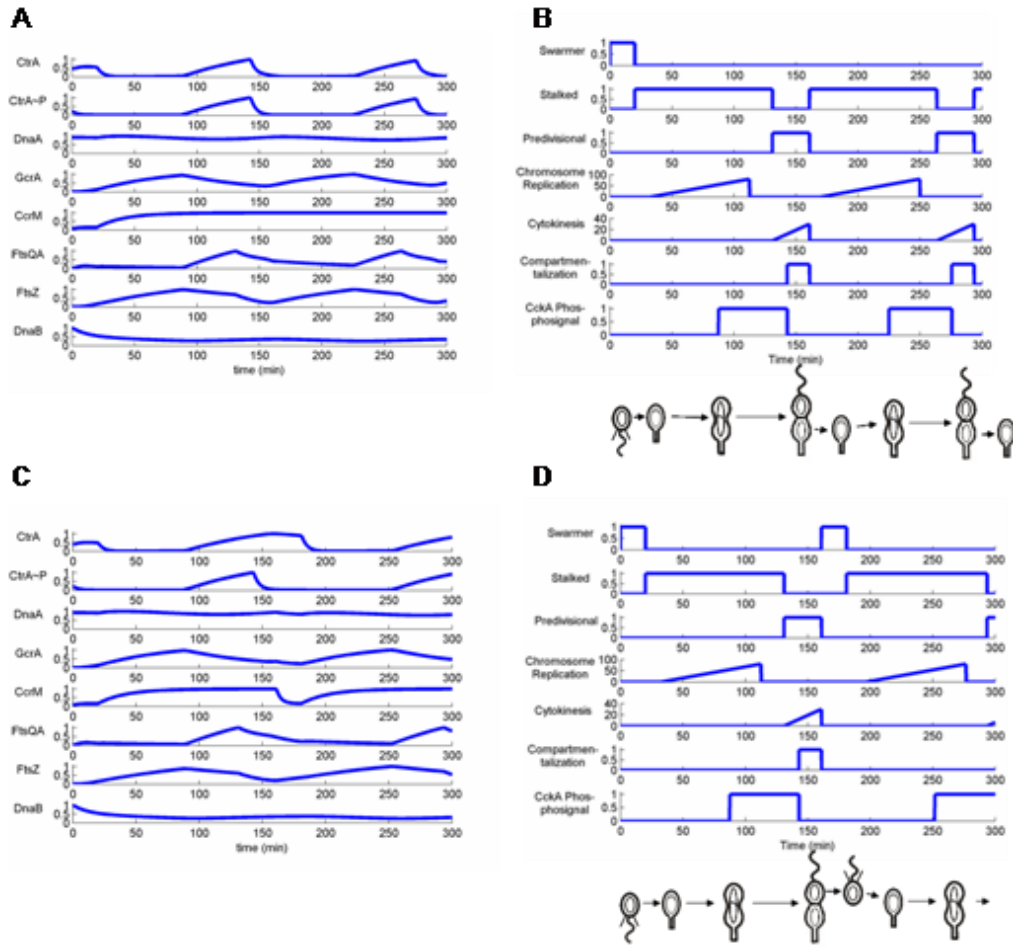


Figure C.3 *In silico* mutant simulation of strain with *ccrM* constitutively expressed. (A) Protein concentration tracking into the stalked compartment. (B) Outputs from Stateflow tracking into the stalked compartment. (C) Protein concentration tracking into the swarmer compartment. (D) Outputs from Stateflow tracking into the swarmer compartment.

3. Permanently phosphorylated and stable CtrA

C. crescentus mutant previously constructed

A mutant strain producing a constitutively active and stable mutant CtrA protein was previously constructed {Domian, 1997 #290}. This strain contains a high copy-number plasmid carrying a mutant *ctrA* gene that encodes a *ctrAD51EΔ3Ω* (pXylX::*ctrAD51EΔ3Ω*) mutant protein. CtrAD51EΔ3Ω is active at all time without phosphorylation, mimicking CtrA~P, and is not subject to proteolysis by ClpXP. These mutant cells showed a dramatic increase in the population size of G1 cells and a corresponding decrease in the population size of G2 cells,

indicating that DNA replication initiation is blocked in most cells. Prior to cell death, cells continued to elongate, forming filaments.

Changes of model parameters to simulate this mutant

To simulate this mutant *in silico*, we used the same equations (Table S2), the same initial concentrations and the same parameters (Table S3A and S3B) as for the wild-type cells, except that “isAlwaysCtrAP” (a binary switch) in Table S3C was set to 1 and “hlCtrAP” (the half-life of CtrA during active proteolysis by ClpXP) was set to a value greater than “hlCtrAc” (the half-life of CtrA without active proteolysis), 200 min in this case.

Results of the mutant simulation

In this mutant simulation, we observed that the levels of CtrA~P increase very fast after the two switches were changed, and then remain high at all times of the cell cycle. Since the levels of CtrA~P are above the maximum threshold to allow the initiation of DNA replication (“cCroCtrA” is 345 nM), the replication of the chromosome is not initiated during the swarmer-to stalked cell transition, even though the levels of DnaA and DnaB are higher than in wild-type cells. We also observed that the levels of FtsZ and FtsQA are higher in the mutant cells than in wild-type cells, but progression of cell constriction is not initiated in the G1 arrested cells, because we have made the assumption that progression of cell constriction is blocked until the replication of the chromosome is complete in our model. Still growing but not able to divide, we predict that the cells will become elongated and filamentous before they die. Hence, this *in silico* mutant simulation agrees with the *in vivo* phenotypes and helps explain what is happening in the cell at a molecular level.

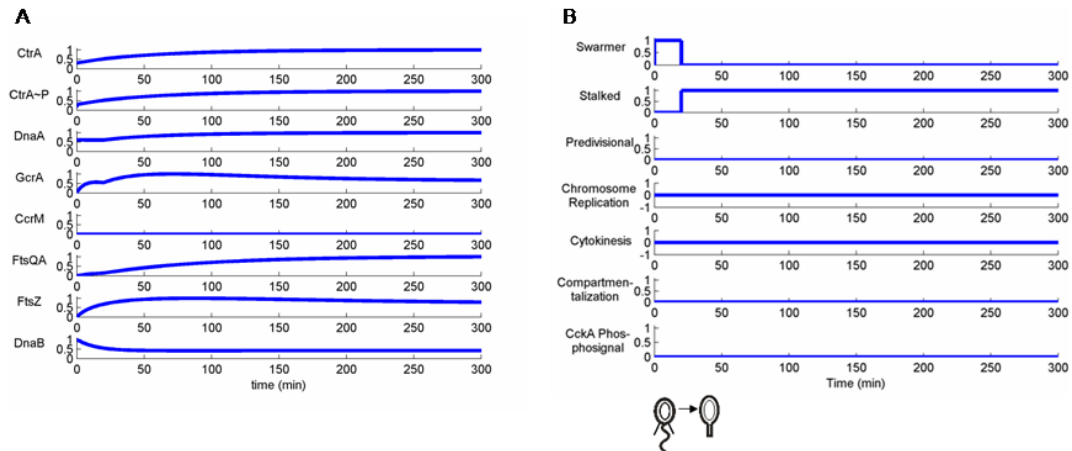


Figure C.4 *In silico* mutant simulation of train with permanently phosphorylated and stable CtrA. (A) Protein concentration timelines (B) Outputs from Stateflow.

4. Move the *ctrA* gene to a chromosomal position next to the terminus

C. crescentus mutant previously constructed

The *ctrA* gene is located at a chromosomal location next to the origin of replication, and its transcription is activated when the *ctrA*P1 promoter becomes hemi-methylated by the passage of the replication fork, soon after the initiation of replication. A mutant strain was constructed {Reisenauer, 2002 #104}, where the *ctrA* gene was moved to a position next to the terminus, and deleted at its native position. The distribution of length of these mutant cells was somewhat broader than the distribution of wild-type cells, and the re-accumulation of CtrA in early pre-divisional cells is delayed for ~15 minutes.

Changes of model parameters to simulate this mutant

To simulate this mutant *in silico*, we used the same equations (Table S2), the same initial protein concentrations and the same parameters (Table S3A and S3B) as for the wild-type cells, except that we changed the value of the “zpc_{ctrA}” parameter (the relative location of the *ctrA* gene on the chromosome) in Table S3C from 0.3 to 1.

Simulation results: Tracking the stalked compartment:

In this mutant simulation, we observed that CtrA~P re-accumulation in pre-divisional cells is delayed by ~15 minutes, because the *ctrA*P1 promoter is kept fully-methylated for a longer time

period of the cell cycle. This result shows that our model is very quantitative, since it simulates the same delay in CtrA re-accumulation as observed *in vivo* {Reisenauer, 2002 #104}. We also observe that the delay in CtrA re-accumulation retards the accumulation of FtsQA, which delays progression of cell constriction by ~15 minutes. Overall, the cell cycle of stalked cells now takes ~125 minutes to complete instead of ~115 minutes for the wild-type stalked cell cycle.

Simulation results: Tracking the swarmer compartment:

We observed that the results of the phenotype simulations are comparable when following both compartments of the cell. Overall, the cell cycle of swarmer cells now takes ~150 minutes to complete instead of ~135 minutes for the wild-type swarmer cell cycle.

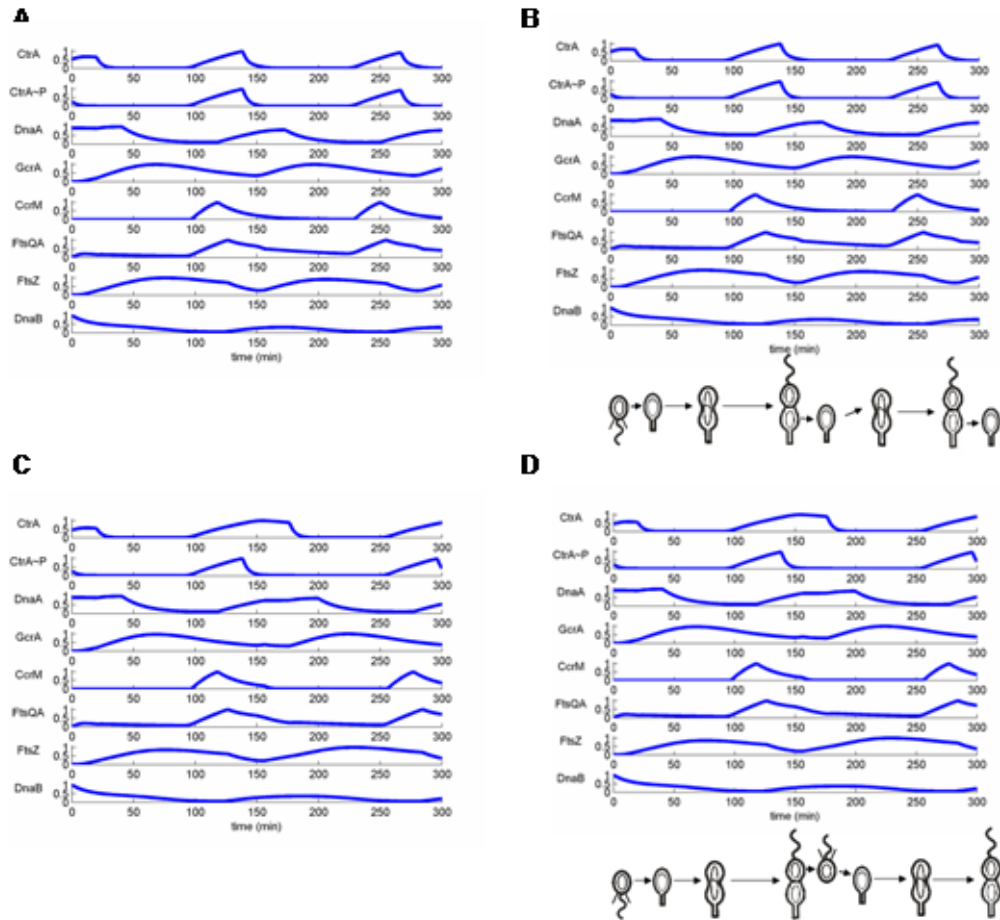


Figure C.3 *In silico* mutant simulation of strain with the *ctrA* gene next to the terminus. (A) Protein concentration tracking into the stalked compartment. (B) Outputs from Stateflow tracking into the stalked compartment. (C) Protein concentration tracking into the swarmer compartment. (D) Outputs from Stateflow tracking into the swarmer compartment.

The commands in Matlab to simulate these *in silico* mutants are listed in <http://www.stanford.edu/group/caulobacter/CellModel>, and the involved parameters are shown in Table C.2.

Table C.2 Parameters for mutant simulations and simulation controls

Symbol	Parameter	Value	Units	Source or rationale
isAlwaysCtrAP	If set to 1, CtrA is always in its active form.	0		It is for mutant simulations that mimic an always active CtrA.
pccrMoe	Synthesis rate of the constitutive promoter driving <i>ccrM</i> expression in mutant simulation	0	nM/s	Set to a non-zero number such as 100 for mutant simulation.
pctrAoe	Synthesis rate of the constitutive promoter driving <i>ctrA</i> expression in mutant simulation	0	nM/s	Set to a non-zero number such as 100 for mutant simulation.
pdnaAoe	Synthesis rate of the constitutive promoter driving <i>dnaA</i> expression in mutant simulation	0	nM/s	Set to a non-zero number such as 100 for mutant simulation.
pgcrAoe	Synthesis rate of the constitutive promoter driving <i>gcrA</i> expression in mutant simulation	0	nM/s	Set to a non-zero number such as 100 for mutant simulation.
pxylose	For mutant simulation, if pxylose is 1, the constitutive promoter is enabled at the time specified by txylose.	0		Provides an extra switch to activate or deactivate the constitutive promoter during a mutant simulation.
t_trackST	At time t_trackST, the simulation switches the type of the cell it tracks according to t_trackST	1		Simulation program parameter
trackST	A binary switch that sets the simulation to track the stalked cell (=1) or the swarmer cell (=0)	0		Simulation program parameter
txylose	Time to induce the constitutive promoter in mutant simulation.	1	min	The value determines the time point when a constitutive promoter is induced in mutant simulations.

Appendix D

Implementation in NuSMV

In the *C. crescentus* model in NuSMV {Cimatti, 2002 #890}, the state variables and update rules for the modular cell cycle functions are organized as a set of individual finite state machines. Some transitions have to wait until specific external conditions (e.g., a regulatory protein reaches a threshold, or a state machine reaches a certain state) are met before other transitions can be executed. There are state machines to model the steps of chromosome replication, the methylation states of certain promoters, stages of cell division, regulatory protein levels, and the phosphorylation state of various proteins.

The NuSMV input files used for the robustness checking of the *C. crescentus* cell cycle control circuit are available online at <http://www.stanford.edu/group/caulobacter/CellModel>. Additional information on the NuSMV implementation is provided in the embedded comments in the NuSMV input files. The README file explains how to run the input file, *Caulobacter.txt*, using NuSMV.

The example below is a simplified version of the chromosome replication state machine in the *C. crescentus* model that illustrates the modeling concepts.

The VAR declaration below defines a state variable that can have one of a list of possible values (which appear in the order that the states progress), according to the "init" and "next" functions specified later. In NuSMV, text after "--" are comments.

```

VAR
chromosome_replication_state : {
  pre_replication,      -- single chromosome before replication starts
  chromosome_rep_init,  -- initiation of chromosome replication
  dnaA,                 -- the replication fork replicates the dnaA gene,
                        -- many states in between. See the full model for
                        -- omitted intermediate states.
  chromosome_rep_end    -- the completion of chromosome replication
                        -- and decatenation of the replicated
                        -- chromosomes
};

```

The ASSIGN declaration defines the actual replication state machine by specifying the initial state values (in the init declaration), and how the state variable is updated on each step (in the next declaration). The next declaration is used to determine the state values for the next step. The state machines in the *C. crescentus* model adhere to a particular convention: immediate transitions for instant reactions are written first, and then there are slow transitions with delays, which will not be evaluated until schedule determines the delay is over.

ASSIGN

```

-- The C. crescentus cell starts in the "pre_replication" state, where
-- there is a single chromosome before cell cycle division starts.
init(chromosome_replication_state) := pre_replication;

next(chromosome_replication_state) := case
  -- The next transition is immediate, since, after the
  -- decatenation of the two replicated chromosomes, the inner
  -- membrane compartmentalization instantly separates them in
  -- two compartments.
  chromosome_replication_state = chromosome_rep_end
    & (cytokinesis_state = Compartmentalization) : pre_replication;

  -- The schedule determines if the delay is over.
  -- Every transition below this must wait for the delay.
  !(schedule = chromosome_replication_delay) : chromosome_replication_state;

  -- chromosome replication is initiated when CtrA_P is low, DnaA
  -- is high, DnaB is high.
  chromosome_replication_state = pre_replication & !CtrA_P & DnaA
    & DnaB
    : chromosome_rep_init;

  -- The dnaA gene is close to the ori.
  chromosome_replication_state = chromosome_rep_init : dnaA;

  -- The replication fork finishes replicating the rest of
  -- of the chromosome.
  chromosome_replication_state = dnaA : chromosome_rep_end

  -- If none of the above conditions is met, stay in the current state.
  1 : chromosome_replication_state;
esac;

```

NuSMV allows nondeterministic assignment, which means that a set of possible alternative values can be specified for a variable. Nondeterministic assignment allows the schedule variable to take on any of the state machines at each step; this enables the model to consider all possible delays.

The protein regulatory network is also modeled as a collection of small state machines, called “regulators,” which correspond to logic gates in computer design. Below is generic NuSMV model for a regulator, which is instantiated for various proteins. When promoter_activity is 1, the promoter of the regulatory gene is activated. After an arbitrary delay, the accumulated protein level becomes 1 (high).

```

MODULE Regulator(promoter_activity, init_level, delay)
VAR level : boolean;
ASSIGN
  init(level) := init_level;
  next(level) :=
    case
      !delay : level;
      1 : promoter_activity;
    esac;

```

The generic gate model is instantiated for the gcrA promoter, which is active when ctrA_P is 0 and DnaA is 1. The protein level GcrA is then asserted after an arbitrary delay included in the gate model.

```

gcrA : Regulator(!CtrA_P & DnaA, 0, (schedule = gcrA_delay));
GcrA := gcrA.level;

```

This is a discrete abstraction of *gcrA* regulation. In a continuous ODE model it would be:

$$\frac{d([GcrA])}{dt} = \frac{\beta/V}{\{1 + (\frac{[C_{d1}]}{[DnaA] + \varepsilon})^n\} \{1 + (\frac{[CtrA \sim P]}{[C_{d2}]})^n\}} - [GcrA] \frac{\ln 2}{hl}$$

Where n , $[C_d]$ and β are Hill function parameters, V is the cell volume, and hl is the half-life of GcrA.

In the NuSMV model for the *C. crescentus* cell cycle control circuit, a dialect of temporal logic called CTL (for "Computation Tree Logic") {Clarke, 1981 #884; Queille, 1982 #885; Emerson, 1985 #896} is used to check whether the cell cycle is completed successfully. For example, the

CTL statement " SPEC AG AF (cytokinesis_state = Divide & (AF cytokinesis_state != Divide));
" states that the cell divides repeatedly under all modeled conditions in perpetuity. More complicated checks were added to check if paths pass the critical cell cycle stages in the right succession order.

Bibliography