

2015

## Variable Data Collection Rate System for a Wildlife Behavior Monitor

Sheldon Blackshire

Follow this and additional works at: <https://researchrepository.wvu.edu/etd>

---

### Recommended Citation

Blackshire, Sheldon, "Variable Data Collection Rate System for a Wildlife Behavior Monitor" (2015). *Graduate Theses, Dissertations, and Problem Reports*. 5219.  
<https://researchrepository.wvu.edu/etd/5219>

This Thesis is protected by copyright and/or related rights. It has been brought to you by the The Research Repository @ WVU with permission from the rights-holder(s). You are free to use this Thesis in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you must obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/ or on the work itself. This Thesis has been accepted for inclusion in WVU Graduate Theses, Dissertations, and Problem Reports collection by an authorized administrator of The Research Repository @ WVU. For more information, please contact [researchrepository@mail.wvu.edu](mailto:researchrepository@mail.wvu.edu).

# Variable Data Collection Rate System for a Wildlife Behavior Monitor

SHELDON BLACKSHIRE

THESIS SUBMITTED TO THE STATLER COLLEGE OF  
ENGINEERING AND MINERAL RESOURCES AT WEST  
VIRGINIA UNIVERSITY IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE DEGREE OF

MASTERS OF SCIENCE IN AEROSPACE ENGINEERING

JAY WILHELM, PH.D., CHAIR  
MARVIN CHENG, PH.D.  
MARIO PERHINSCHI, PH.D.

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

MORGANTOWN, WEST VIRGINIA

July 2015

KEYWORDS: TRACKING DEVICE, PID, GAIN SCHEDULING, DYNAMIC DATA  
COLLECTION

COPYRIGHT 2015

# ABSTRACT

## Variable Data Collection Rate System for a Wildlife Behavior Monitor

Sheldon Blackshire

Behavior monitors typically collect data, and consequently spend energy, at fixed intervals. For devices that utilize energy harvesting, a fixed data collection interval may result in inefficient battery usage due to variability in available solar radiation. Work was performed for a system capable of adjusting a data collection rate, proportional to changes in battery charge, such that data obtained was maximized without sacrificing battery energy sustainability. Energy consumption, of an actual behavior monitor, was modeled to aid in design and evaluation of a changeable data collection rate system. Model validation was performed by comparing simulated to empirical data for battery charge over time. Proportional Integral Derivative (PID) control was used that changed the rate at which data was collected such that error was minimized between battery State Of Charge (SOC) and a reference point. Gain scheduling was incorporated as a mechanism to resist change in data collection rate caused by fluctuation in available SOC. Gain parameters for a discrete, time domain, PID controller were tuned using a manual, trial and error method. Results of tuning showed improved performance with the absence of Integral control. The system was evaluated by performing simulations for change in available solar energy. Results showed that data collection adjusted to changes in available energy and as a consequence, SOC remained within  $\pm 5\%$  of a reference point.

## *Acknowledgements*

I would like to first thank my advisor Dr. Wilhelm for his guidance, mentor-ship, and patience throughout my graduate career. I would also like to thank Michael Lanzone for his financial support so that I could pursue a graduate degree. I would like to thank my sister Amy for her assistance in proofreading. Finally I would like to thank my wife Kelsi for encouraging me through many long nights of writing.

Respectfully,

Sheldon Blackshire

# Contents

<b>Abstract</b>	ii
<b>Acknowledgements</b>	iii
<b>Contents</b>	iv
<b>List of Figures</b>	viii
<b>List of Tables</b>	x
<b>Abbreviations</b>	xi
<b>Symbols</b>	xii
<b>1 Problem Statement</b>	<b>1</b>
1.1 Introduction . . . . .	1
1.1.1 Behavior Monitor . . . . .	1
1.1.2 Energy Constraints . . . . .	1
1.2 Problem Statement . . . . .	2
1.3 Method . . . . .	2
<b>2 Literature Review</b>	<b>4</b>
2.1 Chapter Overview . . . . .	4
2.2 Wildlife Behavior Monitoring . . . . .	4
2.2.1 Constraints . . . . .	5
2.2.2 Sensing Techniques . . . . .	5
2.2.2.1 Radio Tracking . . . . .	6
2.2.2.2 Light Level . . . . .	6
2.2.2.3 Global Positioning Systems . . . . .	7
2.2.3 Classification of Behavior Monitors Based on Data Retrieval . . . . .	8
2.2.3.1 Same-Message Transmitter . . . . .	8
2.2.3.2 Data Loggers . . . . .	9
2.2.3.3 Data Transceiver . . . . .	10
Example 1 . . . . .	11
Example 2 . . . . .	11
2.2.4 Behavior Monitor Selection . . . . .	12
2.2.5 Behavior Monitoring Conclusion . . . . .	12
2.3 System Development . . . . .	13
2.3.1 Data Collection Techniques . . . . .	13

2.3.1.1	Constant Rate Data Collection . . . . .	13
2.3.1.2	Data Collection based on Activity . . . . .	14
2.3.1.3	Dynamic data collection based on Energy Neutral Operation . . . . .	14
2.3.1.4	Summary . . . . .	15
2.3.2	PID Control for Energy Equilibrium . . . . .	16
2.3.2.1	PID Description . . . . .	16
2.3.2.2	Application of PID to Data Collection Rate . . . . .	17
2.3.2.3	Requirement Fulfillment . . . . .	18
	Requirement 1 . . . . .	18
	Requirement 2 . . . . .	18
	Requirement 3 . . . . .	18
2.3.3	Improving Data Collection Rate Consistency . . . . .	19
2.3.3.1	Moving Average . . . . .	20
2.3.3.2	Gain Scheduling . . . . .	20
2.3.4	System Conclusion . . . . .	21
2.4	Modeling . . . . .	21
2.4.1	Battery State of Charge . . . . .	22
2.4.2	Model Summary . . . . .	23
2.5	Chapter Conclusion . . . . .	23
<b>3</b>	<b>Energy Consumption Model</b> . . . . .	<b>25</b>
3.1	Chapter Overview . . . . .	25
3.2	Device Selected to Model Energy . . . . .	25
3.2.1	Hardware . . . . .	25
3.2.2	Software . . . . .	27
3.2.3	Component Selection for Modeling . . . . .	27
3.3	Cumulative Energy Model . . . . .	28
3.3.1	Model Overview . . . . .	28
3.3.2	GPS . . . . .	29
	3.3.2.1 Current Draw . . . . .	30
	3.3.2.2 Timing . . . . .	30
	3.3.2.3 Example . . . . .	31
3.3.3	Microcontroller and Real Time Clock . . . . .	33
3.3.4	Photo-Voltaic Cell . . . . .	33
3.3.5	Battery . . . . .	36
3.3.6	Simulation of Fixed Data Collection Rate . . . . .	37
3.3.7	Model Summary . . . . .	38
3.4	Model Verification . . . . .	39
3.4.1	Evaluation Criterion . . . . .	39
	Relating Battery Voltage to SOC . . . . .	39
	Operating Condition Model Implementation . . . . .	41
3.4.2	Validation Method . . . . .	41
3.4.3	Validation Results . . . . .	41
3.5	Chapter Conclusion . . . . .	43
<b>4</b>	<b>System Development</b> . . . . .	<b>44</b>
4.1	Chapter Overview . . . . .	44

4.2	Controller Design	44
4.2.1	Overview	44
4.2.2	PID Control	45
4.2.3	Gain Scheduler	46
4.3	Controller Gain Tuning	48
4.3.1	Tuning Methodology Overview	48
4.3.2	Constant PVC model for a single controller	49
4.3.2.1	Proportional Gain	49
4.3.2.2	Derivative Gain	49
4.3.2.3	Integral Gain	51
4.3.2.4	Result	52
4.3.3	Variable PVC model for a single controller	53
4.3.3.1	Proportional Gain	53
4.3.3.2	Derivative Gain	54
4.3.3.3	Increased Battery capacity	55
4.3.4	Variable PVC model for dual controllers	56
4.3.4.1	Proportional Gain	56
4.3.4.2	Derivative Gain	57
4.3.5	Tuning Results	58
4.4	Controller Design Summary	59
4.5	System Evaluation	61
4.5.1	Decrease in Available Harvested Energy	61
4.5.2	Increase in Available Harvested Energy	62
4.5.3	Summary	63
4.6	Disturbance Handling	64
4.6.1	Cellular Disturbance	64
4.6.2	GPS Disturbance	65
4.6.3	Disturbance Summary	66
4.7	Chapter Conclusion	66
<b>5</b>	<b>Conclusion and Future Work</b>	<b>68</b>
5.1	Thesis Summary	68
5.1.1	Problem Overview	68
5.1.2	Energy Consumption Model	68
5.1.3	System Design and Evaluation	69
5.2	Discussion	70
5.2.1	Scientific Contribution	70
5.2.2	Future Work	71
<b>A</b>	<b>Simulation Script</b>	<b>72</b>
<b>B</b>	<b>Hardware Classes</b>	<b>80</b>
B.1	Global Positioning System	80
B.2	Photo Voltaic Cell	86
B.3	Battery	91

<i>Contents</i>	vii
<b>C Controller Script</b>	<b>96</b>
<b>Bibliography</b>	<b>98</b>



# List of Figures

2.1	Block diagram for PID feedback control. . . . .	17
2.2	Block diagram of potential implementation of PID. . . . .	18
2.3	Current generation profile (Left: Continuous, Right: Diurnal overlapped) [1] . . .	19
3.1	Schematic overview of CTT-1100 behavior monitor hardware components. . . . .	26
3.2	Current consumption for behavior monitor hardware components. . . . .	27
3.3	Block Diagram for determining current entering or leaving a battery. . . . .	29
3.4	GPS energy consumption model. . . . .	32
3.5	Curve fit for empirical PVC data . . . . .	34
3.6	Variable PVC current generation model. . . . .	35
3.7	Plot (a) shows PVC modeled current. Plot (b) shows SOC for various GPS fix intervals. . . . .	38
3.8	Relationship between battery voltage and SOC . . . . .	40
3.9	Plot (A) Empirical PVC current data. Plot (B) Simulated and empirical SOC. Plot (C) Absolute percent difference between simulated and empirical SOC. . . .	42
4.1	Diagram for an overview of system implementation. . . . .	45
4.2	Block diagram of a PID controller. . . . .	46
4.3	Block diagram for selecting PID gain values. . . . .	47
4.4	Plot (a) shows SOC with respect to time. Plot (b) shows the GPS fix interval. .	49
4.5	Plot (a) shows SOC with respect to time. Plot (b) shows the GPS fix interval. .	50
4.6	Plot (a) shows SOC with respect to time. Plot (b) shows the GPS fix interval. .	51
4.7	Plot (a) shows SOC with respect to time. Plot (b) shows the GPS fix interval. .	52
4.8	Plot (a) shows the variable PVC model. Plot(b) shows battery SOC. Plot(c) shows GPS fix interval. . . . .	53
4.9	Plot (a) shows the variable PVC model. Plot(b) shows battery SOC. Plot(c) shows GPS fix interval. . . . .	54
4.10	Plot (a) shows modeled PVC current. Plot(b) shows battery SOC. Plot(c) shows GPS fix interval. . . . .	55
4.11	Plot (a) shows the variable PVC model. Plot(b) shows battery SOC. Plot(c) shows GPS fix interval. . . . .	56
4.12	Plot (a) shows the variable PVC model. Plot(b) shows battery SOC. Plot(c) shows GPS fix interval. . . . .	57
4.13	Plot (a) shows the variable PVC model. Plot (b) shows battery SOC. Plot (c) shows GPS fix interval. . . . .	59
4.14	Plot (a) shows the modeled PVC current. Plot (b) shows SOC. Plot (c) shows GPS fix interval. . . . .	62
4.15	Plot (a) shows the modeled PVC current for each day. Plot(b) shows SOC with respect to time. Plot(c) shows the GPS fix interval over time. . . . .	63

4.16 Disturbance in battery SOC. . . . .	65
4.17 Disturbance in time necessary to acquire a GPS fix. . . . .	66

# List of Tables

3.1	GPS model properties. . . . .	32
3.2	Table of coefficients for voltage conversion equation . . . . .	40
3.3	Mean Absolute Percent Difference between simulated and empirical SOC. . . . .	43
4.1	Gain values for a dual controller system. . . . .	58

# Abbreviations

<b>SOC</b>	<b>S</b> tate <b>O</b> f <b>C</b> harge
<b>GPS</b>	<b>G</b> lobal <b>P</b> ositioning <b>S</b> ystem
<b>PVC</b>	<b>P</b> hoto- <b>V</b> oltaic <b>C</b> ell

# Symbols

$\epsilon$	Error Signal	SOC %
$u(k)$	Manipulated Variable	-
$PV(k)$	Process Variable	
$S_P$	Control Set Point	SOC
$K_P$	Proportional Gain	-
$K_I$	Derivative Gain	-
$K_D$	Integral Gain	-
$Q_k$	Current Battery Charge	mA-hr
$Q_{max}$	Maximum Battery Capacity	mA-hr
$T$	Sample Period	s
$P_C$	Energy Consumed	mA
$P_S$	Energy Harvested	mA
$B_0$	Energy Storage Buffer	mA-hr
$I$	Current	mA
$E$	Energy	J
$I_{Noise}$	PVC Noise Current	mA
$I_{NET}$	Net Current	mA
$I_{PV}$	Current produced by PVC	mA
$I_{GPS}$	Current consumed by GPS	mA
$I_{MCU}$	Current consumed by MCU	mA
$MAPD$	Mean Absolute Percent Difference	%
$A$	Actual Value	SOC
$S$	Simulated Value	SOC

# Chapter 1

## Problem Statement

### 1.1 Introduction

#### 1.1.1 Behavior Monitor

Wildlife behavior monitors are sensor systems used by biologists for collecting data relevant to animal activity and remotely studying free roaming animals. Migration, foraging patterns, and physiology can be examined using sensor data for purposes such as conservation, environmental monitoring, and ecological behavior. Animal-wearable behavior monitors generally fall under three categories: (a) data loggers, (b) same message transmitters, and (c) data transceivers. Data loggers are lightweight devices that collect and store sensor data to be manually retrieved at a later time by bird recapture or time release triggers where the device detaches itself from the animal. Same message devices wirelessly transmit identification information that requires many base stations to identify behavior. Base stations are electronic devices capable of wirelessly receiving data collected from one or more behavior monitors. Behavior monitors can be equipped with wireless data transceivers that allow two way communication over long distances and are advantageous for immediate retrieval of collected data.

#### 1.1.2 Energy Constraints

Battery powered behavior monitors have limited potential for collecting data based on a finite amount of stored energy. Energy harvesting is commonly used to capture energy from an external source, such as solar radiation, through use of a Photo-Voltaic Cell (PVC). Data

collection potential could be extended as a result of replenishing previously-exhausted battery charge through energy harvesting techniques. One option is to increase battery size proportional to data collection needs, except many devices are designed to minimize weight since a load (>5% body mass) inhibits movement and natural behaviors such as foraging and reproduction [2]. Instead of ever increasing battery capacities, energy harvesting could be used to satisfy weight requirements, without sacrificing available energy for data collection, due to continuous solar radiation being harvested from the environment.

## 1.2 Problem Statement

Behavior monitors typically collect data, and consequently spend energy, at fixed intervals. Devices that utilize energy harvesting with fixed data collection intervals may result in inefficient battery usage due to variability in available solar radiation. Operation with energy harvesting and battery storage may occur in three different ways: (1) Battery charge may become depleted over time due to energy consumed by data collection being greater than energy harvested. Data collection must then stop until enough energy has been harvested to continue. (2) Energy may be harvested faster than it can be consumed (at a given data collection rate). Solar energy typically goes unharvested for a fully charged battery, therefore opportunities for collecting additional data are missed. (3) Energy spent on data collection may be balanced with energy harvested and therefore battery depletion doesn't occur and missed data collection opportunities are minimized.

Balance of harvested and consumed energy was identified as the preferred method of operating a behavior monitor. Operation of a balanced system requires adapting collection rate based on available solar energy therefore, a system was sought to control a data collection rate, proportional to changes in battery charge, to maximize data obtained without sacrificing battery energy sustainability.

## 1.3 Method

Research is presented for developing a system that could be capable of changing a data collection rate, for a behavior monitor, based on available energy. Potential methods for system development, as well as the type of behavior monitor that would most benefit from a changeable

data collection rate, will be evaluated in a literature review. Effects of modifying data collection on battery charge will be simulated by a software model of a behavior monitor. Energy consumption will be modeled based on GPS data collection, energy harvested, and microcontroller operation. Empirical data from a field deployed behavior will be used as input to the model for validation of accuracy. GPS data collection rate and energy harvested would be input to the model and battery charge will be compared to determine model accuracy. Battery charge will be used as input to a controller and will return an adjustment to data collection rate such that battery charge is maintained at a selected point. In addition, gain scheduling may be necessary to maintain a consistent data collection rate that may otherwise vary due to small fluctuation in battery charge.



## Chapter 2

# Literature Review

### 2.1 Chapter Overview

Literature is presented that is relevant to development of a system capable of adjusting a data collection rate, proportional to changes in battery charge, for a wildlife behavior monitor. Wildlife tracking, its relevance, limitations, and techniques are investigated followed by selection of a type of behavior monitor that could most benefit from a variable data collection rate. Techniques that could be used for the system are evaluated based on a set of design requirements. Methods for modeling a behavior monitor for system development are presented, and the chapter concludes with a summary of findings.

### 2.2 Wildlife Behavior Monitoring

Behavior monitoring is over-viewed to obtain fundamental knowledge of how electronic tracking devices are used, their strengths, and limitations. Many methods exist for obtaining and retrieving wildlife data therefore some of the major techniques will be evaluated to determine which could most benefit from a system that controls a data collection rate. The study of wildlife behavior has been beneficial for identifying causes of a species population decline [3]. Factors that influence population dynamics such as habitat modification were studied by tracking wildlife movement [4, 5]. Reduction in size and mass of electronic components have increased the popularity of using animal wearable devices for studying behavior [6]. Embedded sensor systems

are commonly used for remotely monitoring wildlife behavior by collecting sensor data such as position, temperature, heart rate, and movement [7, 8].

### 2.2.1 Constraints

According to literature, two major limitations of behavior monitors exist: size and energy. Wildlife behavior monitors vary in size and can be equipped with sensor systems tailored to meet research objectives. Size and weight of most behavior monitors make placing them on some species detrimental to the animal's movement. Researchers suggest, as a rule of thumb, that behavior monitors should be no greater than 3% to 5% of the bird's weight [2]. According to [8] the majority of all bird species are under 30 grams, therefore only behavior monitors less than 1.5 grams could be used without causing injury to the animal. Overall device weight comes primarily from its battery [9] therefore lighter batteries are often chosen, resulting in further energy constraint.

Unlike continuously powered systems, behavior monitors use batteries and therefore have limited energy due to storage limitations [10, 11]. Due to finite battery capacity, a trade off exists between the rate at which data can be collected and the amount of time a device can remain powered [12]. When battery supply has been exhausted, no additional data can be collected. Replacing batteries for a wildlife behavior monitor may not be practical as it would require animal recapture [13]. Energy harvesting is a commonly used method for extending operational life by providing device power and battery charging [10, 11, 13–18]. Many behavior monitors incorporate photo-voltaic cells that convert solar radiation to energy for battery charging [6, 19–21]. Indefinite amounts of energy could be harvested for powering sensors [15], however only so much can be stored by a battery at a given time [13]. Behavior monitors typically collect data at a conservative fixed interval to ensure a device remains powered [6, 7, 19, 22, 23] and as a result, energy that could have been harvested may be forgone due to storage limitation. Data collection can be performed by many different sensors, where typically the collection rate is selectable.

### 2.2.2 Sensing Techniques

Electronic sensor systems have aided in wildlife monitoring by allowing 24 hour observation, long term data collection over months and years, and reduction in researcher bias due to interference

with an animal during study [24]. Sensors used for data collection vary based on researcher's need, size, and cost. Three commonly used methods of data collection are used: radio tracking, light level, and GPS.

### **2.2.2.1 Radio Tracking**

Radio tracking can be used to electronically study wildlife, where an animal-worn radio transmitter is typically worn that emits a signal which can be interpreted by one or more receivers to determine animal location and activity [25–27]. Radio transmitters can be less than a gram and have been used to study location, heart rate, wing-beat frequency, and respiration [28]. Transmitters typically consist of a power supply, radio, antenna, and optionally a microcontroller. Radios can emit a signal periodically, or in bursts, that contain information necessary for animal identification [29]. Radio transmitter signals can be coded to include sensor data, such as temperature, by altering the signal pulse interval relative to a measured sensor value [24]. Similarly, radio receivers can differentiate between multiple transmitters based on frequency. For example, by assigning each radio transmitter a unique frequency between a range of 148 to 220 MHz, where each device differs by at least 10 kHz, a transmitter can be identified based on the radio frequency alone [30].

Use of a radio receiver is necessary for retrieving signal data related to animal location and behavior. Transmitted signals can be received and interpreted manually by using radio demodulation [24, 26], or automatically where each signal is logged to be retrieved at a later time [29, 31]. Receivers determine location by presence or absence of the RF signal, and in some cases measure signal strength [31, 32]. Depending on the type of receiver (automatic or manual), these devices can include an antenna, radio receiver, amplifier, and data logger. Three commonly used methods for obtaining transmitter location: proximity [24], homing [30, 32], and triangulation [27, 30].

### **2.2.2.2 Light Level**

Light level sensing, also known as solar geolocation, is a method for determining animal position based on light intensity. Solar geolocation uses a light intensity sensor and a real time clock to acquire the time of dawn, dusk, and solar noon [33]. Intensity of solar radiation and the corresponding day of year could be used with solar navigational equations to determine position

in terms of longitude and latitude [34]. Solar geolocation may be susceptible to inaccurately measuring time of dusk when measured light intensity varies from an expected value such as in forests or canyons. Location inaccuracy can be minimized through calibration of light intensity thresholds, such as dusk and dawn, for areas where tracking will occur [35]. Position data was acquired from solar geolocators placed in fixed locations within an 800 km wide location in Western Europe, as well as on native non-migratory songbirds [36]. Data was collected over 12 months, where an average position error was determined to be  $201 \pm 43$  km in latitude and  $12 \pm 3$  km in longitude. Error in position could have been caused by shading, animal behavior, moving to an area where a pre-calibrated threshold no longer applies, or low variation in day length around the equinoxes. While accuracy of light intensity can be questioned, they are currently the only devices capable of tracking small birds on a continental scale [37].

### **2.2.2.3 Global Positioning Systems**

Global Positioning Systems (GPS) are popularly used in wildlife behavior monitoring for studying foraging [6], migration [19, 23], and general movement behavior [7]. According to [38], GPS has benefited the study of ecology in five major ways: (1) It has improved understanding of wildlife habitats and conservation, (2) provided insight into mechanisms of migration, (3) allowed observation for difficult to study species, (4) provided feedback of animal response to human disturbance, and (5) projects impact of climate change. Satellite navigation is used by a GPS receiver to determine position, altitude, speed, and time to name a few [39]. 2/3D position of a globally located receiver can be calculated by acquiring distance measurements from orbital satellites. Distance from a receiver to a satellite can be estimated by measuring signal travel time [40]. Using trilateration [41], distance measurements from multiple satellites can be used to calculate receiver position accurately within 5 meters [42]. Power requirements are a primary constraint for GPS tracking. According to [42] a GPS position fix can require approximately 0.15 watts over 30 seconds (Time necessary to determine position), which places constraint on the amount of GPS data that can be collected given limited available energy. Due to energy constraint associated with GPS receiver operation, most behavior monitors conserve battery life by turning GPS off when not collecting data [24].

### 2.2.3 Classification of Behavior Monitors Based on Data Retrieval

Behavior monitors are designed differently based on many variables such as length of time a device will be deployed, cost, ease of data retrieval, and animal size. Classes of wildlife behavior monitors are introduced based on the method used for data retrieval [8]: loggers, same-message transmitters, and data transceivers. Each class is introduced, pros and cons are evaluated, and an assessment is made for which device classification would most benefit from a changeable data collection rate system.

#### 2.2.3.1 Same-Message Transmitter

Same-message transmitters are devices that use a radio to continuously broadcast identification information, but require one or more receivers to obtain information about animal behavior [9, 26]. Some same-message transmitters weigh as little as 0.2 grams and can be used for many species including insects and small birds [27, 29, 43]. Limited transmission range, operational life, and additional human labor are disadvantages of same message transmitters [24, 31, 44]. Researchers use receiver units with hand-held antennas to collect signals sent by the transmitters. Detection can be automated using base station receivers equipped with rotating or arrays of antennas [27]. The number of receivers needed for acquiring transmitter signal depends on the method of processing data. Receivers are typically used in three ways: Proximity detection, homing, and triangulation.

Proximity detection is the process where a receiver scans a range of frequencies that are used by a wildlife transmitter. Scans can be performed for signal transmission and information of animal presence or absence can be logged [45]. By placing the receiver near places of interest such as nests or known foraging areas, animal proximity with respect to places of interest can be studied [24]. Only one receiver is typically required and retrieving transmitter location can be automated, however position of a transmitter can only be determined to be within proximity of the receiver.

Homing is the process where a user manually locates an animal of interest by receiving transmitted signals by rotating a Yagi antenna until a maximum signal level is detected [24, 30]. Signal strength is constantly monitored as the researcher moves in the direction corresponding to the maximum signal strength until the animal is seen, or until signal strength is at a desired level

[32]. Two disadvantages of homing are human labor costs associated with locating an animal and the introduction of bias because the animal may be disturbed by the researcher.

Triangulation is a method for determining position of a same-message transmitter where the direction of a transmission signal is acquired by a receiver from two or more locations [29, 30]. The intersection of these directions from each receiver location corresponds to the position of the transmitter. Triangulation consists one or more researchers locating a signal direction using an antenna from different positions [9]. Position error can be created during measurement if the animal moves between measurements. Automatic triangulation can be performed using fixed receiver stations with rotating or multiple antennas [27].

Same-message transmitters provide a low complexity method for wildlife tracking. Transmitters are typically smaller and less expensive than other behavior monitors, however receivers must also be factored into the overall cost. Transmission power required for same-message transmitters allow lightweight batteries to be used, but at the expense of transmitter lifetime. Same-message transmitters require researcher labor for manual triangulation and homing, however automatic receivers exist at additional upfront cost. Range of signal transmission is a major limitation, therefore additional receivers may be needed to monitor animals that migrate or have large foraging areas.

### **2.2.3.2 Data Loggers**

Logging devices collect and store data to be retried at a later time [8, 36, 46]. Many of these devices are designed to minimize size and weight such that animals can be tracked without inhibiting their natural behavior [37]. Solar geolocation data loggers, which approximate position by measuring light intensity of sunrise/set, have recently been developed for birds weighing less than 30 grams [46]. Data loggers (sometimes called archivers) typically have no method for transmitting information, therefore unknown data recovery time or a complete loss of data due to the device never being found exists [24, 47]. One study retrieved 26.3% of the tagged birds [35] and 61.8% in another [48]. Some collars are equipped with automatic detachment mechanisms that make the collar fall off after a period of time then a radio transmitter emits a signal to assist researchers in recovering the device [24]. Solar geolocation and global positioning systems are two commonly used position sensors for data loggers [8, 37, 46, 47].

GPS data was archived to study social structure and movement of 14 Parrot's in the Southern Alps over 11 days [47]. The GPS loggers consisted of a commercially available GPS receiver powered by a 380 mA-hr, 3.7V lithium ion battery, which was encased within two layers of heat-shrink wrap to prevent damage from weather or attacks. Each was configured to continuously record position fixes over a 24 hour period at intervals of 1 fix every 3 minutes. The devices and harness weight ranged from 1.9 to 2.6% of the selected Parrot's body weight. Twelve of the 14 loggers were recovered from the field testing site with recapture times ranging from one hour to five days. Periods of activity and rest were clearly identifiable and unusual flight habits were observed. GPS loggers were shown to be viable for tracking Parrot's due to acquiring an average of 55 fixes per day however, a downside of loggers is that they must first be retrieved.

Data loggers are advantageous for monitoring small animals and are cheaper than devices with radio transceivers [8]. Some disadvantages of data loggers are they can be prone to inaccuracy [35], have finite battery life without energy harvesting, and typically require animal recapture to retrieve data [47]. Due to size and weight restriction, data communication such as short range radio or cellular transmission, are typically not present on loggers. Researchers must recapture an animal to acquire sensor data from a logging behavior monitor.

### **2.2.3.3 Data Transceiver**

Data transceivers are an extension of loggers [24, 49] with the ability to transmit collected data, receive program updates, or change device properties through radio communication [50]. Ping transceivers are logging devices equipped with short range radios capable of uploading collected data wirelessly to stationary [6] or mobile [51] base stations within close proximity. Transceivers with short range radios typically rely on close proximity ( $< 10$  km) to a base stations for data transmission [8]. When foreknowledge of an wildlife's migration/roosting area's are not known, behavior monitors are typically designed to hold data for weeks or months before encountering a base station [20].

Data transceivers can be equipped with cellular modems [19, 23, 50] that allow two way communication through a cellular network infrastructure [52]. Data transceivers are advantageous for shortening time between data collection and retrieval [19], performing firmware updates, and monitoring device operation such as data collection rate, battery health, and energy [50]. Communication may require more power consumption than other transceivers (up to 1W peak)

and typically require additional batteries capacity or energy harvesting [19, 24]. Two examples of wildlife data transceivers are presented as follows:

**Example 1** GPS was utilized on a wildlife data transceiver to acquire location data for the Lesser Black-backed Sea Gull (bird) [6]. Body movements were monitored using an accelerometer and environmental monitoring was performed using temperature and barometric pressure sensors. Solar harvested energy was used to charge a battery and power the behavior monitor. Search signals, known as pings, were emitted by a radio transmitter to search for base stations. Multiple base stations were deployed in a network, each having a maximum radio receiver range of 8.5 km. If a base station was within range, a response was sent to initiate communication, transfer collected data, and make modifications to the rate at which data was collected.

**Example 2** CraneTracker was created to monitor the endangered Whooping Crane, which migrates 4000 km annually between Texas and Canada and is capable of traveling 900 km per day [19]. Hybrid architecture was used for transmitting sensor data that relied on cellular networks during migration, and short range radio to base stations for breeding and nesting grounds. Scheduled tasks were as follows: the device woke up, attempted to collect a GPS fix, gathered 10 compass samples over 10 seconds, and then attempted to communicate with both the radio and GSM, before sleeping for 4 hours. CraneTracker was designed for operational life of 7 years by incorporating solar energy harvesting for battery charging. Preliminary testing showed for GPS fixes acquired, 81% had a position error less than 25 meters. For field deployed devices, 94% of all GSM check-in delays were less than 24 hours which signified that GSM technology was capable of reliably transmitting tracking data.

Data transceivers are typically used for reducing time necessary for a researcher to obtain collected data at a cost of increased power consumption. Operational life could potentially be extended by using some form of energy harvesting. Some cellular based data transceivers have been developed that weigh 35 grams [8, 53]. Behavior monitors that use VHF/UHF for data transmission can weigh 15 grams [6, 8], which is critical since an estimated 66% of mammals and 81% of birds were too small to be tracked by the smallest GPS based devices [54].



### 2.2.4 Behavior Monitor Selection

Of the three types of behavior monitors (Data loggers, Burst-message transmitters, and data transceivers), the data transceiver was selected for the design of a variable data collection rate system because of its ability to harvest energy and transmit collected data using cellular networks. Power consumption from the monitoring device can sometimes be negated by harvesting energy from the environment. Data can be transmitted to a researcher through use of a modem. Conservative data collection rates are often selected to ensure the battery remains charged. Development of a system capable of adjusting the data collection rate to prevent battery depletion could benefit data transceivers in three ways: (1) data can be collected at an adaptable rate, (2) time to complete battery depletion is extended, and (3) a light-weight battery may be used because power consumption would be more tightly regulated.

### 2.2.5 Behavior Monitoring Conclusion

Behavior monitors were introduced for studying wildlife for reasons such as conservation, impacts of climate change, and habitat modification. Radio tracking, solar geolocation, and GPS were determined to be three popular methods of obtaining animal behavior information. Analysis of literature revealed that behavior monitors could be categorized in three ways: Data loggers, same-message transmitters, and data transceivers. Behavior monitors were researched to determine what device classification would most benefit from a system that modified data collection based on available energy. Data logging devices reduce weight and power consumption by not using wireless data transmission, but at the expense of manual data retrieval. Due to absence of energy harvesting, conservative data collection rates are commonly used to maximize operational life of data loggers.

Same-message transmitters have low complexity, size, and cost, but suffer from limited transmission range, require significant labor to acquire data, and trade operational life for size. Transceivers typically use GPS to acquire fine-scale location data and can provide rapid feedback for animal behavior using wireless modems, all at the cost of additional power consumption. Energy harvesting can typically be used to extend operational life of a data transceiver but their size makes using them on some animals impossible.

The data transceiver classification was selected because it was considered to most benefit from changeable data collection rate system. Battery charge may be maintained at a safe level by

matching the rate at which data is collected with the amount of energy being harvested. Overall device weight could potentially be reduced by a combination of substituting a smaller battery and more efficiently utilizing available energy.

## 2.3 System Development

Techniques are presented that may be used for development of a variable data collection rate system as well as three requirements to achieve energy equilibrium: (1) Maintain a fully charged battery. (2) Solar energy that goes unharvested due to a fully charged battery should be minimized. (3) Data collection rate should be resistant to change. Data collection methods commonly used by behavior monitors, as well as dynamic techniques are presented. Finally, methods for collecting data at consistent intervals are shown.

### 2.3.1 Data Collection Techniques

Methods for collecting data at a variable rate are reviewed, as well as for maintaining a consistent data collection rate. Techniques were compared and evaluated to determine which may be the most suitable for a variable data collection rate system.

#### 2.3.1.1 Constant Rate Data Collection

One of the most common methods for managing data collection is to use a conservative, fixed collection rate based on expected available energy [6, 7, 19, 22, 23]. If a device needs to perform for a short time such as days or weeks, a fixed data collection rate can be determined based on battery size, deployment time, expected solar energy, and average energy consumption for sensor measurements [50]. Other devices that will be deployed for extended periods of time (years) may collect data only a few times a day because data resolution is less critical than longevity [19]. Some devices can have their data collection rate modified remotely via wireless modem based on researcher observation of battery level. Fixed data collection is advantageous because, when set conservatively, typically results in battery charge remaining full; however, it may result in activities of interest being missed when they occur between measurements [19].

### 2.3.1.2 Data Collection based on Activity

Activity based data collection reduces redundant data collected as a means to conserve battery energy [21, 55]. For sensors that contribute to a large portion of a device energy budget, a reduction in unnecessary sensor use can prolong battery life [20]. Two examples of activity based data collection are accelerometer-informed telemetry and Camazotz.

Accelerometer-informed telemetry used an accelerometer to reduce power consumption, of a behavior monitor, by minimizing redundant GPS measurements associated with animal inactivity [21]. GPS data collection switched between one of three rates (5 min, 15 min, and 60 min) based on the number of sequential accelerometer measurements that were above or below an activity threshold. Compared against a device with a fixed 15 minute GPS data collection rate, activity based sensing attempted 73.6% more locations per day and made 67.4% fewer redundant location attempts for inactive animals.

Camazotz, a multi-modal, activity-based localization method, was developed for studying fruit bats [20]. Behaviors of interest were detected by using low power consumption sensors, relative to GPS. When an event of interest was detected, such as flying, defecation, urination, grooming, or rest, a GPS location fix was acquired. Activity-based sensing reduced power consumption of periodic GPS and single-sensor triggered GPS by up to 77% and 14% respectively while also providing a method to collect GPS data for specific behaviors. Reduced energy consumption could be possible with activity-based localization however, battery depletion could occur during periods of high animal movement or detected behaviors.

### 2.3.1.3 Dynamic data collection based on Energy Neutral Operation

The concept of energy neutral operation has been used in developing methods to dynamically determine data collection rates based energy equilibrium [10, 15, 18, 56, 57]. Energy neutral operation was first defined by [15]. Eq. 2.1 shows the conditions for achieving energy neutral operation for a system with energy harvesting and an ideal energy buffer.  $B_0$  is the initial energy stored in an ideal with no charge leakage or inefficiency. Power consumed  $P_C(t)$  by the device at a specific point in time can be integrated over a time window  $W$  to determine overall energy consumption. Similarly, power harvested by the device  $P_S(t)$  can also be integrated to determine the overall energy generated. Energy neutral operation was originally defined as overall energy consumed being less than or equal to energy harvested plus initially stored energy

however, a more general definition is that energy consumed should not exceed energy harvested [56].

$$\int_0^W P_c(t) dt \leq \int_0^W P_S(t) dt + B_0 \quad \forall W \in [0, \infty) \quad (2.1)$$

One of the first methods for meeting energy neutral operation adjusted data collection based on a prediction model of harvested solar energy [15]. Deviation from predicted and actual energy harvested may result in a sub-optimal data collection rate, leading to periods of depleted or unused battery potential. In addition, computational overhead associated with a solar prediction model is typically undesirable as resources for an embedded system are already constrained. Another method used an adaptive control algorithm to modify data collection, based on available battery energy, to meet the following metrics: energy neutral operation, performance maximization, and duty cycle stability [56]. Selecting a data collection rate that balanced energy consumed and harvested was formatted as a linear-quadratic tracking problem where control laws were developed to minimize the difference between battery charge and a target value. Through use of a moving average, variance in a sensor duty cycle could be reduced to 5%. Finally, Proportional, Integral, and Derivative (PID) feedback control has been shown to make adjustments to the duration of time a device was in a low power state based on the magnitude of a lighting condition and as a result, energy equilibrium was obtained [57].

Dynamic data collection techniques are advantageous because of their ability to adapt to changes in available energy, which could increase device life. Complexity of accurately estimating battery charge due to factors such as temperature sensitivity, rate of charge/discharge, and battery age make applying techniques such as PID and adaptive control challenging [58, 59].

#### **2.3.1.4 Summary**

Research was presented on methods for collecting data that could be useful for developing a system capable of changing a data collection rate based on available energy. Fixed data collection rates are commonly used by wildlife behavior monitors. Based on the unpredictable nature of energy harvested, a fixed data collection rate could result in a sub-optimal amount of data collected. Some behavior monitors collect data based on animal activity, however battery life may be reduced for active animals. Dynamic data collection, based on energy neutral operation, shows promise in achieving energy equilibrium for variable data collection rate system. Early

work used prediction models of energy harvested that were prone to inaccuracy to determine a data collection rate. However, recent work used techniques that implicitly account for energy harvested by measuring available battery energy. Use of battery energy as a controlled parameter could be complex due to several factors associated with accurately estimating charge.

### 2.3.2 PID Control for Energy Equilibrium

Proportional, Integral, Derivative (PID) control is introduced as a potential method for modifying a data collection rate of a behavior monitor to achieve energy equilibrium. Methods of using PID as the mechanism for varying a data collection rate are presented and the feasibility of using PID is evaluated for monitoring a consistent data collection rate.

#### 2.3.2.1 PID Description

PID is a control technique commonly used within industrial controller systems [60, 61]. Control of a desired parameter is achieved by first generating an error function  $\epsilon(k)$ , where  $k$  represents discrete sampling. As seen in Eq. 2.2,  $\epsilon(k)$  was a difference in a desired set point  $S_P$  and process variable  $PV(k)$ .

$$\epsilon(k) = S_P - PV(k) \quad (2.2)$$

The objective of a PID controller is to minimize an error function, thus driving a process variable to a desired state [62]. Figure 2.1 shows a block diagram for a discrete, time-domain PID controller that applies proportional, integral, and derivative action to  $\epsilon(k)$  in order to generate a change in a control variable  $u(k)$ . Proportional action was applied by multiplying a gain  $K_P$  by the current error  $\epsilon(k)$ . The use of Integral action reduces a steady state error by adjusting  $u(k)$  based on a summation of  $\epsilon$  past and scaling the sum by an integral gain  $K_I$  and the time between successive samples  $T$ . Derivative action contributes to  $u(k)$  based on a prediction of future error. Inspection of the rate of change of  $\epsilon$  could be accomplished by using a backward difference between current  $\epsilon(k)$  and previous  $\epsilon(k-1)$  error over the sampling time  $T$ , which could then be scaled by a derivative gain  $K_D$  [63].

$$u(k) = K_P\epsilon(k) + K_I T \sum_{i=0}^k \epsilon(i) + K_D \frac{\epsilon(k) - \epsilon(k-1)}{T} \quad (2.3)$$

The variable being controlled  $u(k)$  can be adjusted at each time step by summation of the proportional, derivative, and integral terms. Each additional call to the PID function requires recalculation of  $u(k)$ . Therefore a change in PV will cause a corresponding change in  $u(k)$  until PV converges to SP.

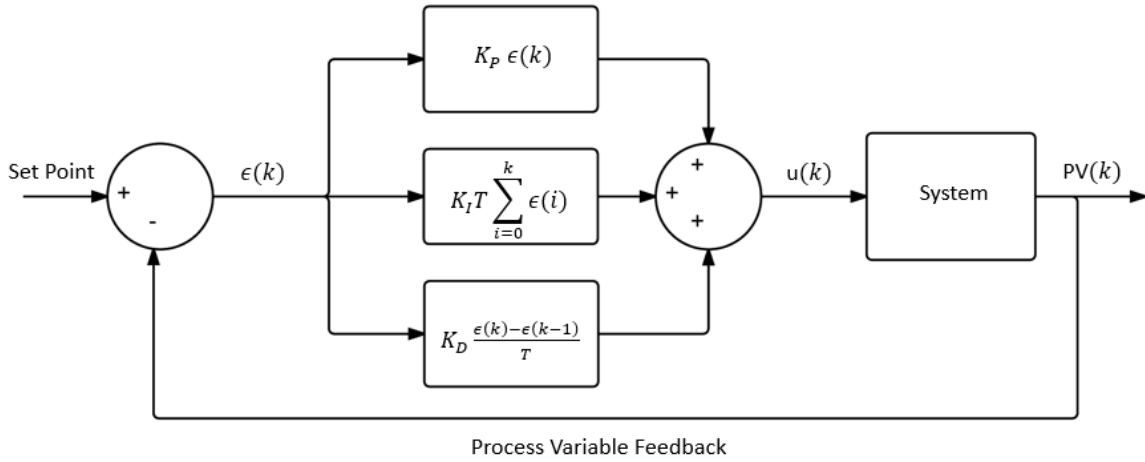


FIGURE 2.1: Block diagram for PID feedback control.

### 2.3.2.2 Application of PID to Data Collection Rate

PID control could be used to develop a variable data collection rate system, for achieving energy neutral operation, by the following method: (1) an error signal  $\epsilon$  (Eq. 2.2) could be generated by taking the difference between battery charge and a designated reference charge. (2) Using Eq. 2.3,  $\epsilon$  could be converted to an adjustment in data collection rate such that future error signals are minimized. (3) The rate at which data collect is collected is adjusted such that battery charge approaches a reference. Periodic repetition of steps (1-3) could result in a battery level approaching a designated reference state with time. Changing in the rate at which data is collected could cause a behavior monitor to experience a corresponding change in battery charge. Figure 2.2 shows a visual illustration of how PID could modify a data collection rate.

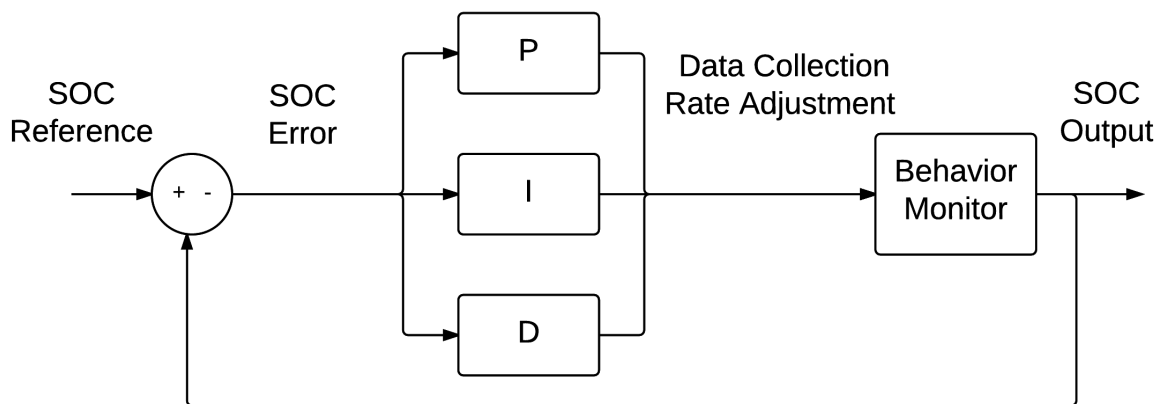


FIGURE 2.2: Block diagram of potential implementation of PID.

### 2.3.2.3 Requirement Fulfillment

PID control could potentially be used in a variable data collection rate system. In order for PID to be viable, the following requirements must be met: (1) Maintain a fully charged battery. (2) Solar energy that goes unharvested due to a fully charged battery should be minimized. (3) Data collection rate should be resistant to change.

**Requirement 1** PID control could use battery charge as the variable to be controlled and data collection rate as the variable to be manipulated. If a decrease in energy harvested occurs, the battery will consequently drop in charge. By assigning a set point for which to maintain battery charge, any change in available energy would result in a change in the controller error function  $\epsilon$ . PID action would then attempt to correct the error by adjusting a data collection rate.

**Requirement 2** Through minimization of an error function, a PID controller could maintain a constant battery charge. In order to ensure that available solar radiation gets harvested, a set point for battery charge could be used that is not close to 100% (fully charged).

**Requirement 3** Based on (1) and (2), any positive or negative fluctuation in energy harvested would result in a corresponding change in the controller error function. Maintaining a consistent data collection rate may not be possible when a PID controller experiences fluctuation in its error function due to control action being applied to the error.

### 2.3.3 Improving Data Collection Rate Consistency

Behavior monitors are often used for studying movement patterns of an animal over extended periods of time. If a researcher is equally interested in animal activity and rest, sampling at a consistent rate may be advantageous over activity based techniques such as [20] or [21]. PID is a promising method for modifying data collection rate based on available energy, however it may be unable to maintain a consistent data collection rate due to a solar diurnal cycle or adverse weather causing fluctuation in the process variable (Battery charge). The solar diurnal cycle is a pattern of solar radiation that recurs every 24 hours based on a full rotation of the earth [15, 64]. Figure 2.3 shows current measurements from a solar cell sampled every 10 seconds for 70 days [1]. The plot on the left shows current over each day, and the right plot shows each day overlapped on a diurnal scale. By inspecting the diurnal plot, it can be seen that current generated by a solar cell varies based on time of day, which may cause a corresponding change in battery charge. Two methods are presented that may be used by a PID controller such that diurnal changes in battery charge don't result in corresponding change in data collection rate.

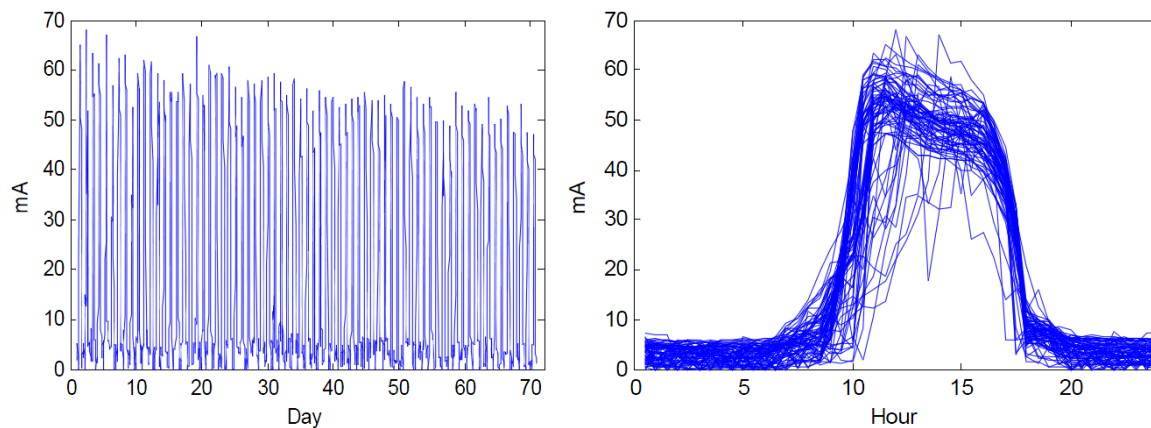


FIGURE 2.3: Current generation profile (Left: Continuous, Right: Diurnal overlapped) [1]



### 2.3.3.1 Moving Average

One method that has been used for smoothing smaller-than-signal noise values is a moving average [16]. Data collection rate  $u(k)$ , as seen in Eq. 2.4, was smoothed to  $\bar{u}_t$  using an exponentially-weighted moving average by summing the previously averaged data collection rate  $\bar{u}_{t-1}$  with  $u(t)$ , scaling the sum by a smoothing parameter  $\alpha$ , then adding to  $\bar{u}_{t-1}$  to acquire  $\bar{u}_t$ .

$$\bar{u}_t = \bar{u}_{t-1} + \alpha(u_t - \bar{u}_{t-1}) \quad (2.4)$$

$\alpha$  could be modified between 0 and 1, where values closer to 0 increased smoothing and values near 1 reduced it. Selection of an optimal  $\alpha$  may not be possible for a wildlife behavior monitor as too much smoothing could result in device failure because data collection rate cannot change fast enough when solar energy available strongly diminishes. Similarly, too little smoothing may cause fluctuation proportional to the solar diurnal cycle. Potential causes of a fluctuating data collection rate include variation in solar radiation or animal relocation to an area with less solar radiation. The exponentially weighted average could be used to smooth a data collection rate, but  $\alpha$  would need to be evaluated such that battery life is prolonged, otherwise a different type of moving average could be used.

### 2.3.3.2 Gain Scheduling

PID is typically used for controlling systems that have linear relationships between the output and control variable [60]. The concept of gain scheduling can be used for nonlinear systems where different sets of PID gains are used at various operating points for the output variable [62, 65]. Gain scheduling can refer to predefined sets of PID gains  $K_P$ ,  $K_I$ , and  $K_D$  being changed based on some condition. Design of a gain scheduled controller was presented in [66] as a four step procedure. (1) Develop a model for the system and identify an input parameter. (2) Establish sets of gains (multiple controllers) for operating points of the parameter. (3) Implement and tune each controller, and (4) test. Gain scheduling could be used with a PID controller such that data collection rate could be rapidly modified when error in battery charge (Eq. 2.2) surpassed a threshold, and resist change while within a threshold. Gain scheduling could also reduce fluctuation in data collection rate, caused by diurnal changes in battery charge,

based on how close battery charge is to a set point (magnitude of the controller error function  $\epsilon$ ). Data collection rate would not be changed as long as  $\epsilon$  remained within a set range of values. One of the major disadvantages of PID gain scheduling is that additional tuning must be performed for each level of operation.

### 2.3.4 System Conclusion

Techniques were investigated for developing a system capable of modifying data collection rate of a transceiver type behavior monitor based on available energy. Three requirements were first established for use in evaluating data collection techniques found in literature: (1) Maintain a fully charged battery. (2) Solar energy that goes unharvested due to a fully charged battery should be minimized. (3) Data collection rate should be resistant to change. An overview of fixed data collection, activity based data collection, and dynamic data collection were presented. Dynamic data collection is typically based on energy neutral operation, a potentially useful basis for system development. The next section outlines a method for modeling a data transceiver type behavior monitor that could be used for system development and testing.

## 2.4 Modeling

Research is presented on energy consumption modeling, of an embedded system, that may be applicable for developing a changeable data collection rate system. Online testing (Testing on an actual device) could be costly and time intensive. Based on [67], systems can be analyzed by analytical methods, computer simulation, and device implementation. Analytical methods could result in inaccuracy due to complexity of device operation and uncertainty of future energy harvested as in [15]. Simulation is the most widely used method for battery powered wireless system analysis due to fast development, and the ability to design and modify algorithms based on simulation feedback [68].

Energy consumption of a wireless sensor node was modeled based on components such as device hardware, solar energy harvesting, and device operating behavior (Such as data collection rate or wireless communication). Device hardware within the model platform included operation of a solar panel, sensors, and supporting electronics, and a battery. In [68] a simulator was developed to simulate hardware, software, energy consumption and the environment, for designing and evaluating algorithms. Inaccurate model assumptions or inability to properly predict solar

radiation are a few reasons model inaccuracy could occur. In order to evaluate model accuracy, [15] used actual solar data to validate an energy prediction model. Developing an energy consumption model could, for a behavior monitor, be used to develop, test, and evaluate performance for a changeable data collection rate system. Due to potential inaccuracy between modeled and actual energy consumption, actual behavior monitor data would be needed to validate the model such as empirical battery level, energy harvested from a solar panel, and the duty cycle of a sensor.

### 2.4.1 Battery State of Charge

Battery charge may be used as a metric for regulating a data collection rate of a behavior monitor using a PID controller. Batteries store electrochemical energy that cannot be directly measured like fuel level remaining in a vehicle [69, 70]. State of Charge (SOC) is a commonly used term for charge estimation, and considerable research has gone into improving its accuracy [59, 69–72]. SOC is defined in Eq. 2.5 as the ratio of charge capacity  $Q$  to the maximum capacity  $Q_{max}$ , then expressed as a percentage from 0-100 corresponding to completely empty or full respectively [73].

$$SOC = \frac{Q}{Q_{max}} * 100 \quad (2.5)$$

Several methods have been developed for indirect estimation of SOC, where two of these methods are Coulomb (charge) counting, and measuring battery terminal voltage. Charge counting estimates SOC (see Eq. 2.6) by summing the charge that enters and leaves a battery. Accumulated charge could be determined by measuring current  $I(k)$  entering or leaving a battery at a discrete sample interval  $\Delta t$ , and then multiplying them together to obtain charge [58, 73]. An adjustment value for SOC could then be acquired by dividing by the maximum charge  $Q_{max}$  that could be held by a battery.

$$SOC(k) = SOC(k - 1) + \frac{\Delta t * I(k)}{Q_{max}} * 100 \quad (2.6)$$

Two main drawbacks of charge counting are the need for an initial SOC measurement to estimate future charge and an accumulation of error that may result from self-discharge. Measuring voltage between battery terminals is another method of estimating SOC. Voltage can be assumed

to decrease linearly as the battery is discharged however, a major limitation is that the battery voltage is also affected by temperature and magnitude of charge/discharge [74]. Relationships between SOC and voltage can be compensated for by through empirically derived maps between voltage and SOC at various discharge rates and temperature [73].

### 2.4.2 Model Summary

Modeling energy consumption could be used for developing and evaluating a changeable data collection rate system for a wildlife behavior monitor. Model construction based on major energy consuming or generating components, and net charge flow could be tracked using a battery. State of Charge is a commonly used indicator for determining the estimating remaining energy in a battery but can be prone to inaccuracy. Inaccuracy based on modeling assumptions could propagate to system design therefore some method of validation may be needed.

## 2.5 Chapter Conclusion

Literature was reviewed to acquire beneficial information to aid in development and testing of a changeable data collection rate system based on available energy. Behavior monitors were researched to determine what device classification would most benefit from a system that modified data collection based on available energy. The data transceiver classification was considered to most benefit from data collection rate system because battery charge could be maintained at a safe level by matching the rate at which data is collected with the amount of energy being harvested. Use of smaller batteries could be possible as a result of more efficient energy management and ultimately resulting in an overall reduction of device weight.

Techniques were investigated for developing a system capable of modifying data collection rate of a transceiver type behavior monitor based on available energy. Three requirements were established for use in evaluating techniques found in literature: (1) Maintain a fully charged battery. (2) Solar energy that goes unharvested due to a fully charged battery should be minimized. (3) Data collection rate should be resistant to change. An overview of fixed data collection, activity based data collection, and dynamic data collection were presented. Dynamic data collection typically uses the principal of energy neutral operation, a potentially useful basis for system development. PID control was shown to fulfill all of the requirements when gain scheduling was used.

Modeling energy consumption could be used for development and evaluation of a changeable data collection rate system based on major energy consuming or generating components, and net charge flow could be tracked using a battery. State of Charge is a commonly used indicator for determining the estimating remaining energy in a battery but can be prone to inaccuracy. Inaccuracy based on modeling assumptions could propagate to system design therefore some method of validation may be needed. The next chapter presents a software model of a wildlife transceiver-class behavior monitor that will be used for system development.

## Chapter 3

# Energy Consumption Model

### 3.1 Chapter Overview

Design of a software model is presented that represents energy consumption, of a wildlife behavior monitor, to be used as a tool for system development and evaluation. The behavior monitor to be modeled is presented as well as a description of how each component was used to form a cumulative model of energy consumption. Evaluation of model accuracy is shown by comparison of simulated energy consumption with empirical data.

### 3.2 Device Selected to Model Energy

Wildlife behavior monitors can be a complex system consisting of many electrical components [19, 20, 22]. The CTT-1100 (Cellular Tracking Technologies, LLC) behavior monitor was used as reference for building a model of energy consumption. Relationships between power and function, as well as operation behavior, and schedules were gathered for modeling major electronic tracking device components that consumed energy.

#### 3.2.1 Hardware

Behavior monitors typically consist of a microcontroller, one or more sensors, flash data storage, an energy source, a data communication device, and optionally an energy harvesting device. Current draw of the CTT-1100's hardware components were measured by placing a Tektronix

4040 digital multimeter in series with a XPOWER 305D DC power supply and the behavior monitor. Current draw of each component's state (Active or sleep) was then measured individually. As seen in Table 3.2, an MSP430F5528 microcontroller (Texas Instruments) drew 2.32 mA while active at 8 MHz, and 2.1 uA while sleeping. The position sensor, an ORG447 Global Positioning System (GPS) receiver, consumed on average 27.5 mA while active and 0.441 mA while sleeping. An RX-8564LC (Epson Toyocom) real time clock was used to determine when to wake the device from sleep. It was always active, and consumed 1.5 uA. Data communication was handled by a GE865-QUAD (Telit) GSM module with average current draw of 825 mA while active. A Photo-Voltaic Cell (PVC) was capable of generating a maximum 150 mA from solar radiation. Current consumed by the CTT-1100, while in a low power state, was measured to be 40 uA. Figure 3.1 shows a general hardware component overview for the CTT-1100.

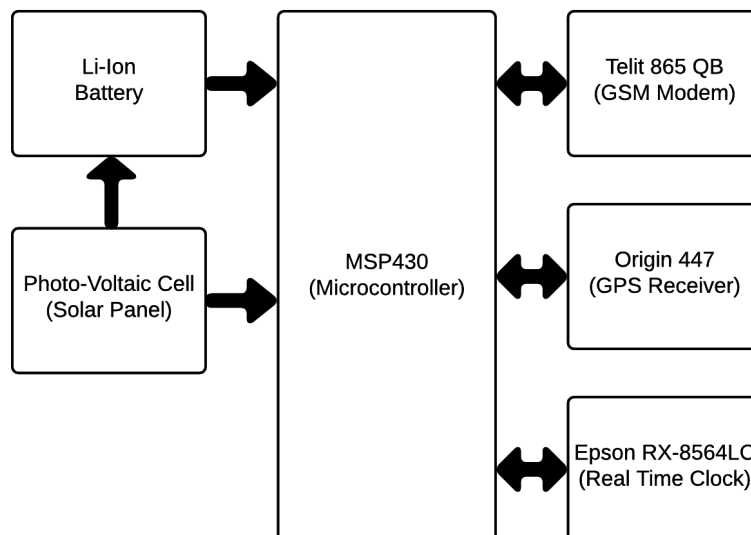
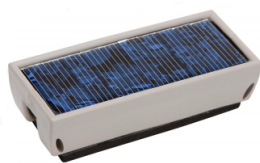


FIGURE 3.1: Schematic overview of CTT-1100 behavior monitor hardware components.



Component	Mode	Current	Units
MSP430F5528	Active	2.32	mA
	Sleep	2.1	uA
Origin 447X	Active	27.5	mA
	Sleep (Fix)	441	uA
Epson RX-8564LC	Active	1.5	uA
Telit 865 QB	Active	825	mA
Photo-Voltaic Cell	Active	0 - 150	mA

FIGURE 3.2: Current consumption for behavior monitor hardware components.

### 3.2.2 Software

In addition to current draw, information related to component operation was obtained for model construction. GPS sensor data were acquired by a CTT-1100 during periods of sunlight and stored it in flash memory. Three data collection intervals were available for use: 1 second, 30 seconds, or 15 minutes. For battery preservation, the microcontroller and GPS were placed in a low power, sleep state when data were not being collected. Data were transmitted to a remote server once per day via cellular network, then removed from flash memory. If the battery dropped below the minimum level (30% SOC), all GPS, and uploading via modem, activity was disabled until the battery was charged to 80%.

### 3.2.3 Component Selection for Modeling

Energy consumption was modeled based on the CTT-1100 behavior monitor introduced in Sections 3.2.1-3.2.2. GPS was used by a behavior monitor to acquire location data based on satellite signals. Solar energy was harvested to power the behavior monitor, as well as charge its battery, therefore a photo-voltaic cell was included in the model. Control and execution of behavior monitor tasks were handled by a microcontroller, and a real time clock (hardware device) was responsible for keeping time. While together the microcontroller and real time clock accounted for only 8% of the current draw for GPS in the active state, however they were modeled because over time their energy consumption may be significant. Energy was drawn from an 800 mA-hr lithium ion battery in the event of solar radiation being insufficient to power the device, therefore a battery was necessary to model. Wireless data communication



was handled by a GSM modem which had an average current draw of 825 mA (while active). Due to the modem only being powered for a few minutes each day, its operation was treated as a disturbance, presented in Section 4.6.

### 3.3 Cumulative Energy Model

Energy consumption for a wildlife behavior monitor was modeled through software. Development tools and programming methodology for model construction are shown and a convention for tracking energy consumption between components is introduced. Finally, a description of each component of the software model is presented followed by a model development summary.

#### 3.3.1 Model Overview

Energy consumption of the CTT-1100 behavior monitor was modeled to be used as a tool for development and performance evaluation for a variable data collection rate system. The model operated by determining charge  $Q(k)$  entering or leaving a modeled battery at discrete sampling times  $k$ . As seen in Eq. 3.1, net current consumption, at time  $k$ , was determined by a cumulative measurement of current draw  $I_i(k)$  for each of the  $N$  components in the model. Charge for each sampled time was then determined by multiplying net current by the time interval  $\Delta t$  between measurements.

$$Q(k) = \Delta t * \sum_{i=1}^N I_i(k) \quad (3.1)$$

The amount of current entering or leaving a battery was determined by models for a global positioning system, photo-voltaic cell, and combination of microcontroller and real time clock. Each of these components were modeled to require a certain amount of current to function for each time step. GPS and the real time clock draw current from the battery, where the battery draws current from the PVC (See Figure 3.3). Current entering or leaving the battery was equated to electric charge (mA-hr) by the amount of time  $\Delta t$  between samples. Cumulative battery charge was positive for cases when solar energy was more than sufficient for powering the device, and negative when it was not. For the first case the battery was considered to be charged, and depleted for the second.

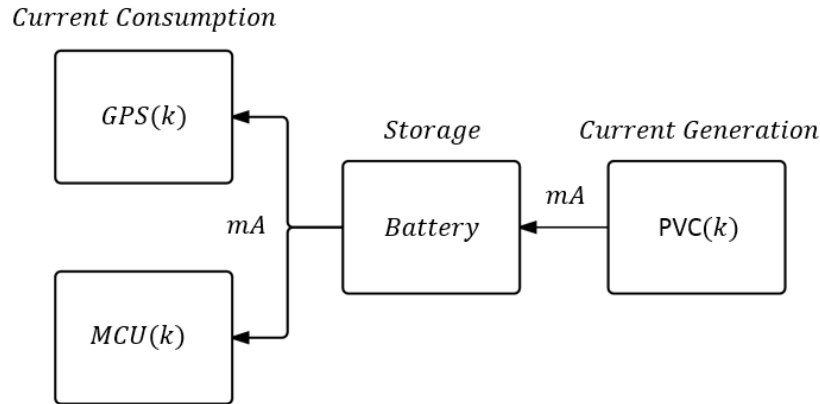


FIGURE 3.3: Block Diagram for determining current entering or leaving a battery.

Empirical modeling is a tool for gaining insight, quantifying, or simulating particular aspects of a system [75]. A software model of energy consumption, for a behavior monitor, was developed using MATLAB 2014 because of its strength in numerical computing and plot tools for viewing data. Each of the major hardware components, introduced in Section 3.2.1, were modeled using Object Oriented Programming (OOP). Software behavior was organized into functions (Example: Energy consumption for GPS based on data collection) and hardware characteristics into properties (Example: current draw and measurement duration). Object properties could be altered in order to better simulate events such as change in solar energy availability. By using OOP, the model could be broken into smaller modules to be substituted or modified without having to modify other software components. The remainder of this section introduces each of the components that make up the cumulative energy consumption model.

### 3.3.2 GPS

GPS was used by a behavior monitor for acquiring wildlife migration including position, velocity, altitude, heading, and time.

Precise orbital information, known as ephemeris data, had to be acquired by a GPS receiver, from a minimum of three satellites, before it could accurately determine its location [40]. Energy consumed by a GPS receiver could not be ignored because time necessary to acquire a measurement could take anywhere from 30 seconds to minutes based on satellite signal strength [9]. This section introduces the methodology behind modeling energy consumption of a GPS receiver based on time to acquire data (fix) and current draw during the active and sleep states.

### 3.3.2.1 Current Draw

Three current draw states were used for the GPS data collection cycle: (1) Active, (2) Sleep, and (3) Off. (1) Active mode was defined as the amount of current drawn by the GPS when a fix was being acquired. (2) Sleep mode was defined as the current drawn while not attempting to collect GPS data. (3) Off mode corresponded to no current being drawn because the device was disabled. The three current draw states were used together to model GPS current draw for the data collection cycle as follows. Active mode was used between the time the device turned on until a fix was acquired. Depending on the amount of time until the next fix attempt, sleep or off mode was then entered. The value used for 'active' mode was 27.5 mA, 'sleep' was 0.441 mA, and 'off' was 0 mA. The next section introduces the amount of time each spent in each state.

### 3.3.2.2 Timing

The time to acquire a location fix was determined by relating GPS fix interval (GFI) to the amount of time between consecutive GPS measurements taken by the behavior monitor. For example, the device introduced in Section 3.2 had three GFI's that could be used: 1 second, 30 seconds, or 15 minutes. Time to First Fix (TTFF) was defined as the amount of time necessary to calculate an accurate location from an initial startup. TTFF was based on factors such as validity of ephemeris data and satellites in view, both of which change depending on the amount of time since a GPS measurement was last taken [76]. Three common classifications exist that can be used for estimating TTFF: (1) Hot start, (2) warm start, and (3) cold start. (1) For a hot start, conditions for calculating location, from the last measurement, haven't changed much and therefore TTFF is small (typically one or two seconds). (2) For a warm start, enough time was passed since the last fix for a satellite to leave view, or information on last known position to become invalid. In this case, ephemeris data must be acquired, for the necessary satellite, which could take up to 30 seconds. (3) For a cold start, the receiver is missing or has inaccurate information regarding its global position, or has lost satellite visibility. In this case, navigation data must be acquired for a minimum of three satellites which could take minutes depending on signal strength. The GPS model was developed by assuming time to acquire a location fix was related to the amount of time between consecutive GPS measurements, or GPS fix interval (GFI). Hot, warm, and cold starts were modeled by defining a rule set that assigned a constant TTFF value based on GFI. For a GFI less than or equal to 10 seconds, TTFF was modeled as

1 second because GPS had accurate satellite information. For GFI less than 15 minutes but greater than 10 seconds, TTFF was modeled as 7 seconds to allow for receiver timing, position, or ephemeris data updates. Finally, if GFI was greater than 15 minutes, or the GPS was being switched on from the Off state, TTFF was modeled to be 60 seconds because ephemeris data for multiple satellites needed to be acquired.

Depending on the amount of time between each fix, energy could be saved by switching the GPS off instead of using sleep mode. Turning the device off would require a cold start (60 second TTFF) for the next GPS measurement, therefore energy spent sleeping had to exceed energy consumed acquiring a fix to justify turning the device off. The amount of energy consumed collecting a GPS measurement was equivalent to  $TTFF$  and the level of current draw  $I_{Active}$ . The amount of energy consumed while sleeping was equivalent to the critical sleep time  $t_{critical}$  for determining if the GPS should be powered off and current consumed while sleeping  $I_{Sleep}$ . By combining these energy consumption quantities, as seen in Equation 3.2,  $t_{critical}$  could be acquired.

$$TTFF * I_{Active} = t_{critical} * I_{Sleep} \quad (3.2)$$

Applying Eq. 2.2.4 to a warm start (TTFF between 10 and 900 seconds),  $t_{critical}$  results showed GPS should be turned off if sleeping for longer than 55 minutes. Because this time exceeded the 15 minutes threshold (between a warm and cold start) for GFI, it was determined that the device should not be switched off for a GFI within the warm start region. Critical sleep time for a cold start was approximately one hour and two minutes, therefore any time GFI was larger than this time, the device should be turned off instead of put in a sleep mode. For a GFI less than 10 seconds, GPS was modeled to be continuously powered to ensure a fix could be acquired before the next scheduled attempt. Current draw for each state, TTFF values, and GFI thresholds are summarized in Table 3.3.

### 3.3.2.3 Example

Simulation of the GPS model was performed over five minutes where GFI was set to 1 minute and current draw was measured every second. GPS energy consumption started in the off state at time zero, and entered active mode in one second, which triggered a cold start GPS fix. GFI was set to 1 minute therefore the device entered sleep mode until the next fix attempt. The

Name	Value	Units
Active Consumption	27.5	mA
Sleep Consumption	0.441	mA
Off Consumption	0	mA
Cold Start GFI	$t > 15$	min
Warm Start GFI	$900 > t > 10$	s
Hot Start GFI	$t \leq 10$	s
Cold Start TTF	60	s
Warm Start TTF	7	s
Hot Start TTF	1	s

TABLE 3.1: GPS model properties.

GPS entered active mode with a TTF of 7 seconds every minute for the remainder of the simulation, which can be seen in Figure 3.4.

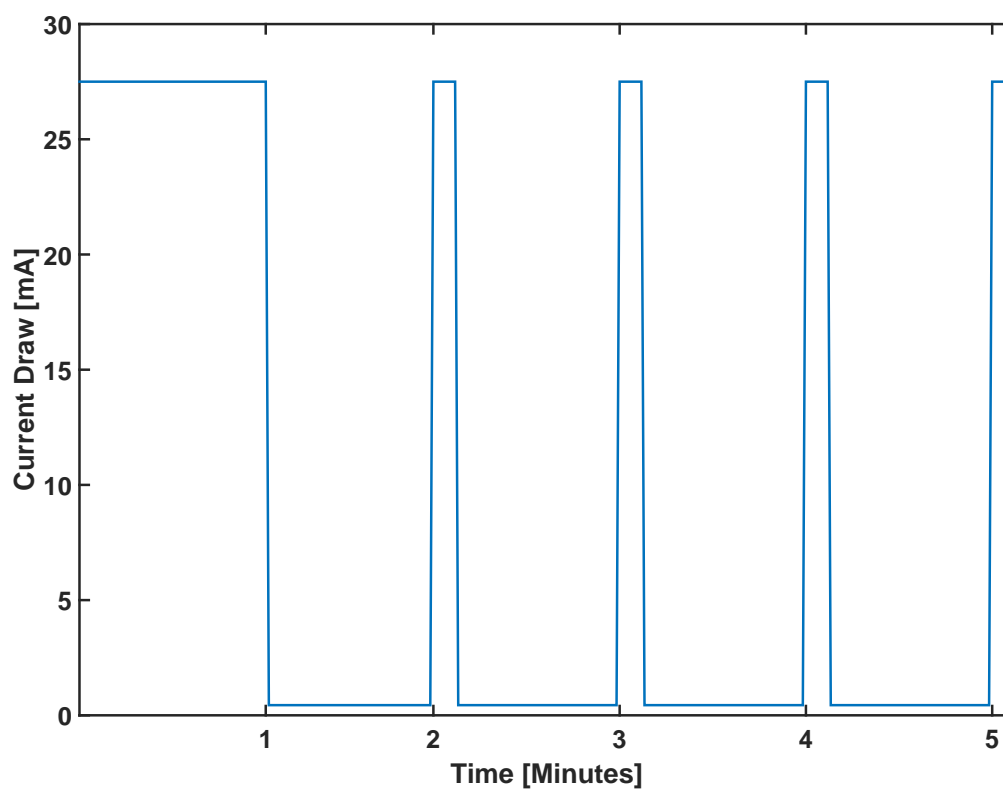


FIGURE 3.4: GPS energy consumption model.

### 3.3.3 Microcontroller and Real Time Clock

Behavior monitor functions such as data collection, communication, sleep, and data storage were facilitated through a microcontroller. Two energy consuming states were available for a microcontroller: Sleep and active. Sleep mode was modeled to have a constant current draw of 2.1  $\mu\text{A}$  and was used when a behavior monitor was not collecting data. Alternatively, active mode was used during data collection with a modeled current draw of 2.32 mA. Accurate system time and the ability to generate interrupts for scheduled events were handled by a real time clock. Providing the battery of a behavior monitor was not completely depleted, a real time clock was modeled to always be active (current draw of 1.5  $\mu\text{A}$ ). Because the microcontroller had two modes of operation behavior (on or sleep), and the real time clock had one state (always on), they were modeled as one component. State was determined by testing for GPS activity. If GPS was collecting a fix, current draw from the active microcontroller state was summed with real time clock current at each sample time. Otherwise, net current draw was modeled as the sum of real time clock active and microcontroller sleep current.

### 3.3.4 Photo-Voltaic Cell

An energy generation model was developed based on a Photo-Voltaic Cell (PVC), which generated power from solar radiation. PVC's are commonly used by behavior monitors to extend operational life by charging the battery, however the rate of current generation can vary significantly based on many factors. An empirical model was developed for energy harvested by a photo-voltaic cell, in the form of current being generated, at each discrete sample time  $I(k)$  of a solar diurnal cycle. Empirical data were examined for current generated by a PVC over a 30 day period, shown in Figure 3.5. Average daily PVC current was calculated from the 30 days of data, a simple moving average was used to smooth noise, then a second order polynomial curve fit was applied, of the form shown in Eq. 3.3. Coefficients  $a$ ,  $b$ , and  $c$  were determined using the MATLAB function polyfit.

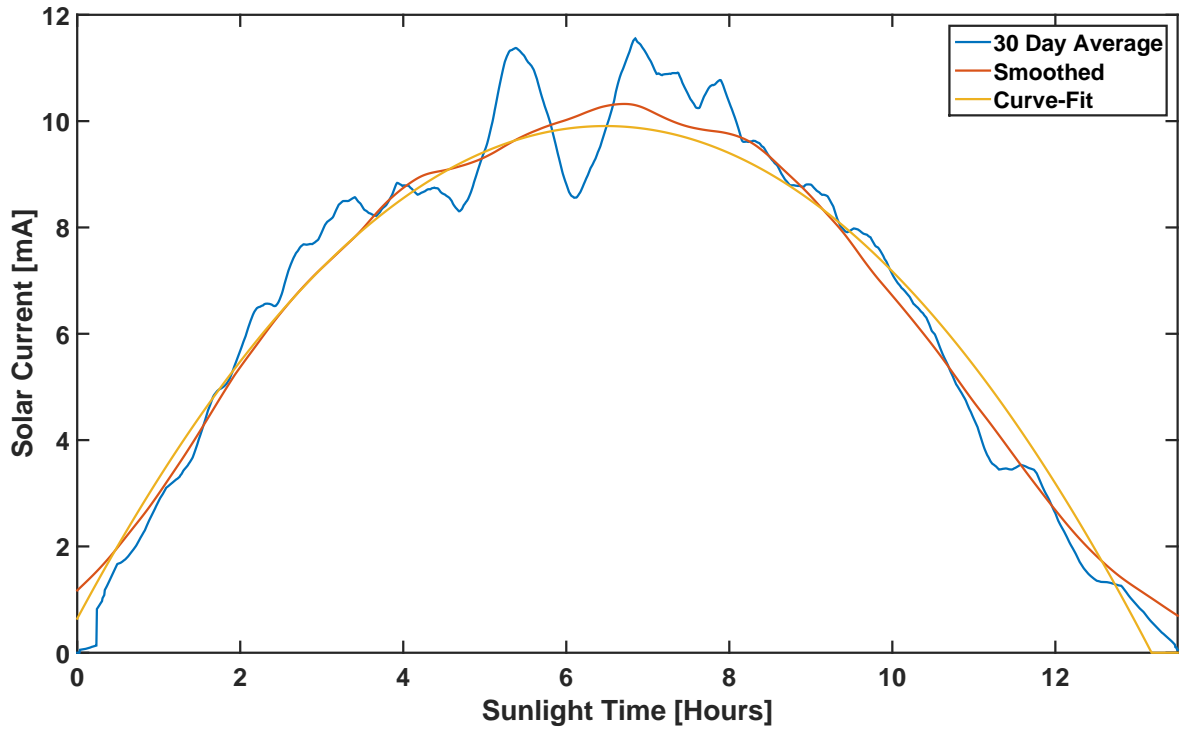


FIGURE 3.5: Curve fit for empirical PVC data

$$I(k) = ak^2 + bk + c \quad (3.3)$$

Based on empirical data from multiple PVC devices, an average of 10.5 hours of sunlight duration occurred for the devices, and the maximum current draw ranged from 10 mA to 40 mA. Energy harvested by a PVC was modeled based on the following parameters: (1) Current generated by a PVC at sunrise, (2) PVC current generated at midday, and (3) PVC current generated at sunset. Daylight was selected to start at 6:00 AM and with a current draw of 0 mA and nighttime was selected to start at 4:30 PM and also had a current draw of 0 mA. Midday occurred at 10:30 PM and had a current generation of 10 mA. The maximum current at midday can be scaled to represent anything from 0-150 mA to model different weather patterns or regions. Selected time and current values were input into the MATLAB function polyfit to generate polynomial coefficients for  $a$ ,  $b$ , and  $c$ . Eq. 3.4 shows the polynomial used for modeling PVC current generated from 6:00 am to 4:30 pm where a maximum current of 10 mA was drawn at 10:30 pm. The maximum current at midday, as well as sunrise and sunset times, could be modified to model different weather patterns, geographic regions, or levels of shade, however this would require updating the polynomial coefficients.

$$I(k) = -2.711 \times 10^{-8}k^2 + 2.213 \times 10^{-3}k + -3.514 \times 10^1 \quad (3.4)$$

Noise component for current generated was introduced to model factors such as cloud cover, partial PVC obstruction, or lack of sunlight due to foliage. This was accomplished by generating random current values  $I_{Noise}$ , at time  $k$ , between a lower  $I_{Min}$  and upper  $I_{Max}$  bound, which represented  $\pm 20\%$  of  $I(k)$  respectively. Incorporation of noise into the PVC model can be seen in Eq. 3.5.

$$I_{Noise}(k) = I_{Min}(k) + (I_{Max}(k) - I_{Min}(k)) * Rand(0, 1) \quad (3.5)$$

Figure 3.6 shows current produced by a PVC model over a period of 24 hours at a time step of 1 second. Times before and after daylight corresponded to zero current generated by the model. Between the sunrise and midday, the value of current increased parabolically until a maximum value was reached. From midday to the beginning night, values for current decreased parabolically until a minimum of 0 mA occurred. Scaling the magnitude of current generated each day is a unique strength of the model because it allows different scenarios for weather or bird behavior to be evaluated, while a potential drawback was the limited impact noise played in determining current generation. During periods of darkness a typical current profile has zero magnitude then current increases with parabolic behavior when the sun comes up. The amount of current generated would plateau when the sun was directly overhead then would begin to decrease until the night.

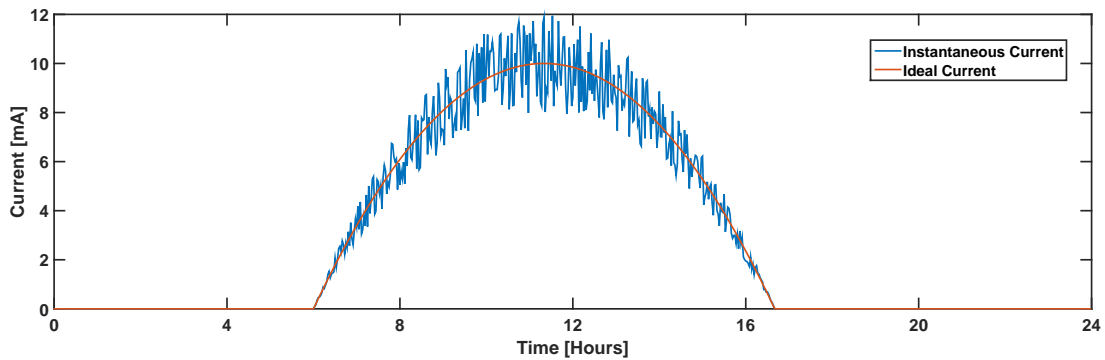


FIGURE 3.6: Variable PVC current generation model.



### 3.3.5 Battery

The CTT-1100 tracking device, selected for modeling, contains a lithium ion battery used when solar current was insufficient and was modeled by considering two properties: Method for energy storage, and flow (charge and discharge). The battery model was developed as a storage container that could hold a minimum and maximum amount of electric charge. Stored battery energy was expressed in milliamper-hours (mA-hr) which describes how much current (mA) can be released from the battery in one hour. Maximum capacity was selected to be a commonly used battery size of 800 mA-hr, however this could be modified based on the application. Level of energy in the battery could not exceed the maximum capacity in the event of charging, and was considered completely discharged when at 0 mA-hr. Amount of energy entering or leaving the modeled battery was measured in discrete time steps. Using Eq. 3.6, net current draw  $I$  required for powering other modeled components (GPS, MCU, RTC, and PVC) was converted to energy by multiplying by the time step, which depending on the sign, could be added or subtracted from the battery.

$$I_{Net}(k) = I_{PV}(k) - I_{GPS}(k) - I_{MCU}(k) \quad (3.6)$$

Lithium ion batteries ideally charge in a two stage process: constant current and saturation charge [77]. First, charging occurs with constant current until approximately 90% (saturation point) charge capacity is reached. The battery then continually accepts less current until max capacity is reached [78]. The first stage of charging was modeled by assuming any positive net current charges the battery. The second stage was modeled by linearly decreasing the amount of charge that can enter the battery as the battery charge level exceeds a saturation point  $Q_{Sat}$ . Eq. 3.7 shows that the current amount of energy stored in the battery  $Q(k)$  was equivalent to the energy from the previous time step  $Q(k-1)$  plus incoming energy  $I(k)\Delta t$  scaled to account for charge saturation.

$$Q(k) = Q(k-1) + I(k)\Delta t \left\{ 1 - \frac{Q(k-1) - Q_{Sat}}{Q_{Max} - Q_{Sat}} \right\} \quad (3.7)$$

Battery State of Charge  $SOC$ , defined as the ratio between current  $Q(k)$  and maximum charge  $Q_{Max}$  capacity of the battery, was used for tracking available energy. Coulomb, or charge counting was used to determined SOC by auditing the amount of current entering or leaving

the battery [73]. SOC was determined at each time step  $SOC(k)$  by dividing the net charge entering or leaving the system  $Q(k)$ , then adding it to the previous value  $SOC(k - 1)$  (See Equation 3.8). For the occurrence of the first calculation of SOC, the battery was assumed to be fully charged therefore having an initial SOC of 100%.

$$SOC(k) = SOC(k - 1) + \frac{Q(k)}{Q_{Max}} * 100 \quad (3.8)$$

### 3.3.6 Simulation of Fixed Data Collection Rate

Simulation of the CTT-1100's energy consumption was performed using constant GPS fix intervals for a constant amount of energy harvested each day. Solar current generated, see Figure 3.6, was used for all 15 days of the simulation such that differences in SOC could be attributed to GPS energy consumption. Three GPS fix intervals were evaluated: 1 second, 30 seconds, and 15 minutes. Based on Plot (b) of Figure 3.7, battery SOC was initially modeled as 100% to represent a fully charged battery. Battery SOC was completely depleted by day 4, for a GPS fix interval of 1 second, so GPS data collection stopped until the battery was charged to 80%. GPS fix interval of 30 seconds showed a linear decrease in SOC over 15 days and 15 minutes showed battery charge maintained at approximately 97%.

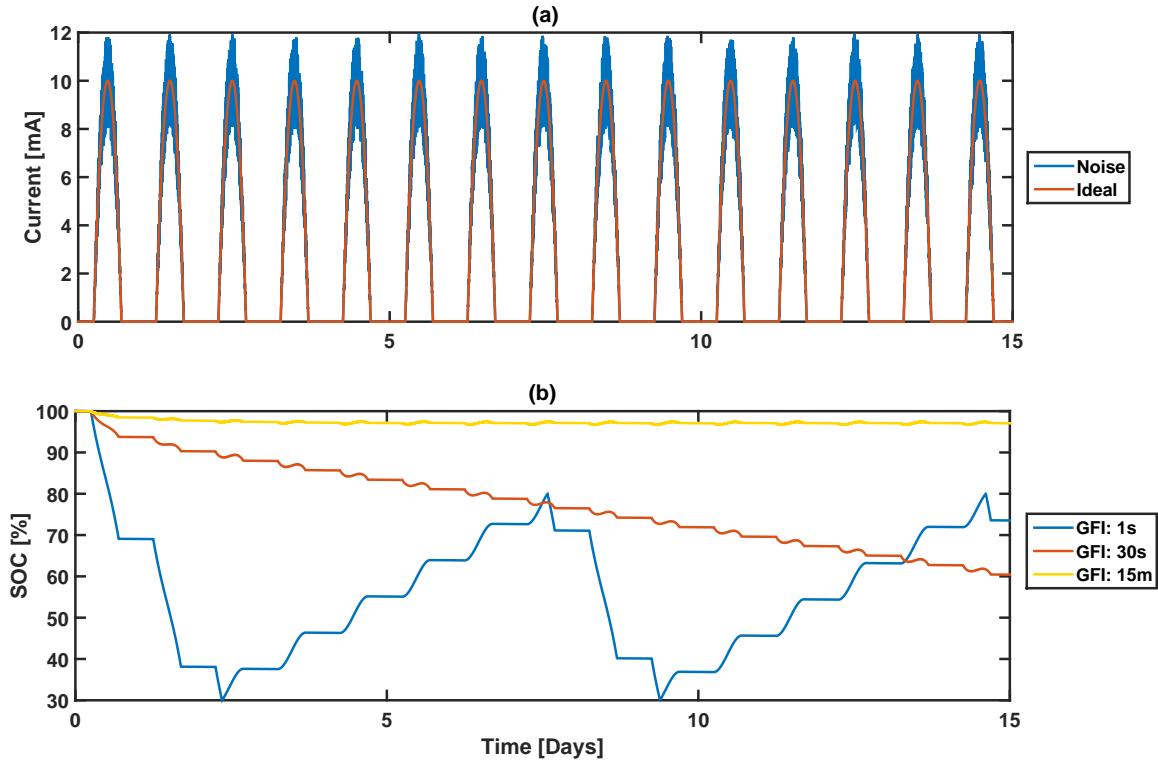


FIGURE 3.7: Plot (a) shows PVC modeled current. Plot (b) shows SOC for various GPS fix intervals.

### 3.3.7 Model Summary

An energy consumption model of a behavior monitor was needed for development and performance prediction of a changeable data collection system. The CTT-1100 tracking device was analyzed to establish a basis for a software model. Hardware and software for the CTT-1100 were analyzed to determine major energy consumption components as well as how they were used. GPS was the largest source of current consumption, using 27.5 mA while acquiring a fix. Time to first fix for GPS was modeled by developing a rule set based on data collection rate. The photo-voltaic cell was modeled for charging a battery using energy harvested from solar radiation. Current generation of a PVC was modeled parabolically in proportion to a solar diurnal cycle. Noise was added to the ideal current generation curve to better simulate energy harvest for adverse weather or PVC obstruction. Simulation of the CTT-1100's energy consumption was performed by summing current consumption from GPS, microcontroller, and real time clock with current generated by a PVC and applying net current to a lithium ion battery model using a 1 second time step. Evaluation of SOC for the three GPS fix intervals

was performed for consistent available solar energy, which showed a system for manipulating data collection would be beneficial for self regulating SOC.

### 3.4 Model Verification

The objective of creating an energy consumption model was for use as a tool in developing and testing a variable data collection rate system. Before design or performance testing, accuracy of an energy consumption model for a wildlife behavior monitor needed to be determined. Model validation is presented by comparing simulated to empirical data.

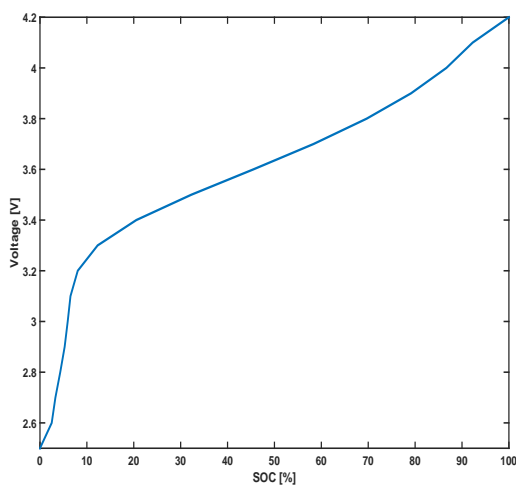
#### 3.4.1 Evaluation Criterion

Validation was performed by adjusting operating parameters for the tracking device energy consumption model to match empirical data acquired from a wildlife behavior monitor. For a given level of operation, a comparison was performed between modeled and empirical derived battery SOC. The following parameters were simulated based on empirical data: (1) The range of time in a day for which GPS data was collected, (2) Initial battery voltage, (3) Current generated by a PVC, and (4) the GPS fix rate.

**Relating Battery Voltage to SOC** The energy consumption model used SOC for determining available battery charge, however empirical data was measured in volts. On the device, voltage measurements of an 800 mA-hr battery were taken at regular intervals for a 0.01 C load. In order to perform a comparison, battery voltage was converted to SOC. A Vencon UBA5 battery tester was used to determine the relationship between voltage and time. The 800 mA-hr battery was tested at a 0.01C load line, which was the same load applied to the tracking device when battery voltage measurements were taken. Battery characterization load (Vencon UBA5) must be equal to the CTT-1100 load when battery measurements were taken, otherwise the relationship between battery voltage and SOC will be inaccurate. The MATLAB function for cumulative trapezoidal integration (`cumtrapz`) was then used to integrate the load current over time, resulting in a relationship between electrical charge and voltage. Given a maximum battery capacity  $Q_{Max} = 800$  mA-hr, electrical charge over time was converted to SOC with the ratio of electrical charge  $Q$  over  $Q_{Max}$  from Eq. 2.5. Using the MATLAB curve-fit toolbox (`cftool`), the SOC vs. Voltage data set was then modeled into an equation that could be used in

simulation through a rational polynomial curve-fit. Eq. 3.9 shows  $SOC(V)$  where  $V$  is the input voltage, and  $p_1 - p_6$  and  $q_1 - q_3$  are polynomial coefficients (Refer to Table 3.2 for coefficient values). The polynomial was subtracted from 100 to represent available instead of used SOC. Eq. 3.9 took a voltage bound between 2.5V and 4.2V and returned a value between 0% and 100% SOC respectively (shown in Figure 3.8).

$$SOC(V) = 100 - \frac{p_1V^5 + p_2V^4 + p_3V^3 + p_4V^2 + p_5V + p_6}{q_1V^2 + q_2V + q_3} \quad (3.9)$$



$P_x$	Value	$Q_x$	Value
$p_1$	28.43	$q_1$	1
$p_2$	-473.1	$q_2$	-6.669
$p_3$	3080	$q_3$	11.33
$p_4$	-9749		
$p_5$	14850		
$p_6$	-8526		

FIGURE 3.8: Relationship between battery voltage and SOC

TABLE 3.2: Table of coefficients for voltage conversion equation

**Operating Condition Model Implementation** Operating conditions used to simulate empirical data were implemented in the model as follows: (1) Solar current generated was linearly interpolated from 15 minutes to match the model sample period of 1 second. The PVC model was then replaced with actual solar current data. (2) The fix interval for the GPS model was set to 15 minutes. (3) Modeled battery capacity was set to 800 mA-hr, and the initial battery SOC was determined by inputting the initial voltage to Eq. 3.9. (4) The simulation duration was configured based on the length of time between the first and last empirical data record for each day.

### 3.4.2 Validation Method

Validation of the CTT-1100 energy consumption model was performed based on a comparison between empirically derived and simulated SOC for one day of data collection. An absolute difference  $|\Delta SOC(k)|$  was used for comparing simulated and empirical battery SOC at each time step (1 second) of a simulation.  $|\Delta SOC(k)|$  was calculated by taking a difference of actual  $SOC(k)_{Actual}$  and simulated  $SOC(k)_{Model}$  SOC then dividing by  $SOC(k)_{Actual}$  as seen in Eq. 3.10.

$$|\Delta SOC(k)| = \left| \frac{SOC(k)_{Actual} - SOC(k)_{Model}}{SOC(k)_{Actual}} \right| * 100 \quad (3.10)$$

Eq. 3.11 shows mean absolute difference measurements were taken by summing  $|\Delta \overline{SOC}|$  over the course the simulation length  $L$ .

$$|\Delta \overline{SOC}| = \frac{1}{L} \sum_{i=1}^L \left| \frac{SOC(i)_{Actual} - SOC(i)_{Model}}{SOC(i)_{Actual}} \right| * 100 \quad (3.11)$$

### 3.4.3 Validation Results

Mean absolute percent difference between simulated and empirical battery SOC was calculated for each of the 31 days in January 2015 for four CTT-1100 behavior monitors. The first behavior monitor was located in Kruger National Park, South Africa during the time data was collected. Figure 3.9 shows a comparison of modeled and empirical SOC (Plot (A)) for actual PVC data (Plot(B)) as well as the corresponding absolute difference (Plot(C)) for a period of GPS data collection. Battery SOC was initially 70% and declined to 68% for hours 11 to 13. A spike in

PVC current of 11 mA occurred at hour 13 that caused SOC to increase to 71%. For the entire day the absolute difference in SOC never exceeded 2%.

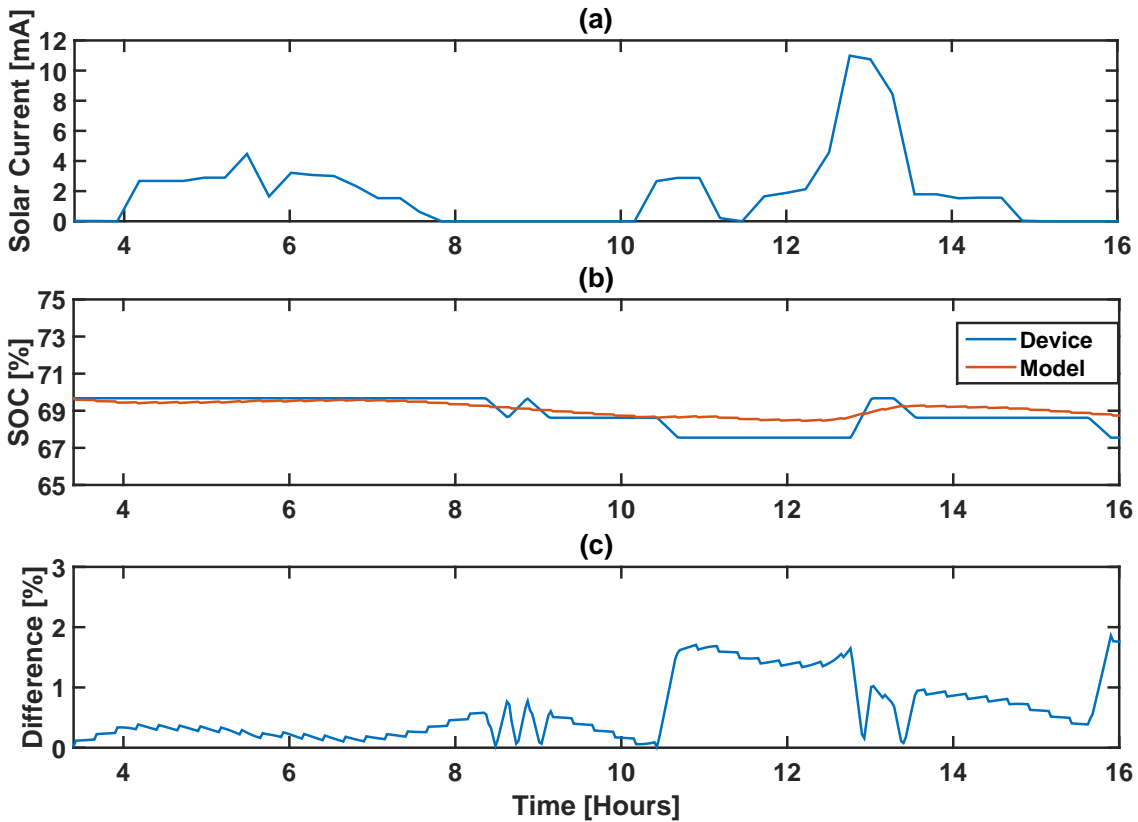


FIGURE 3.9: Plot (A) Empirical PVC current data. Plot (B) Simulated and empirical SOC. Plot (C) Absolute percent difference between simulated and empirical SOC.

Comparison between simulated SOC and actual showed an average MAPD of 3% over 31 days where the minimum and maximum were 0.6% and 12% respectively. The second behavior monitor was also located in Kruger National Park, South Africa, with an average MAPD of 5%, minimum of 1% and maximum of 13%. The remaining devices were located in Michigan, United States during the time of data collection. Comparison between simulated SOC and actual showed an average MAPD of 1% over 31 days where the minimum and maximum were 0.3% and 2% respectively. The second behavior monitor was also located in Michigan, United States with an average MAPD of 1%, minimum of 0.1% and maximum of 5%. The results of MAPD can be seen in Table 3.3. Based on results of the energy consumption model's level of accuracy, a maximum average error of 5% was deemed acceptable for use in a changeable data collection rate system.

Device Location	Absolute Difference		
	Minimum	Maximum	Average
South Africa	0.6%	12%	3%
South Africa	1 %	13%	5%
Northeastern United States	0.3%	2%	1%
Northeastern United States	0.1%	5%	1%

TABLE 3.3: Mean Absolute Percent Difference between simulated and empirical SOC.

### 3.5 Chapter Conclusion

Energy consumption of a wildlife behavior monitor was modeled for development and evaluation of a changeable data collection rate system. Energy consumption was modeled based on the CTT-1100 tracking device, where a GPS, PVC, microcontroller, and real time clock cumulatively dictated available battery energy. Software architecture was analyzed for developing behaviors for each component of the model. Energy consumption of a GPS receiver was modeled based on a rule set that related time to acquire a measurement to the data collection rate. PVC current generated from solar energy harvesting was modeled proportionally to a solar diurnal cycle. Random noise was also incorporated and the magnitude of energy harvested could be scaled differently for each day. Energy consumption for a microcontroller was represented by an active and sleep component and a real time clock applied a constant level of consumption. Finally, a lithium ion battery was modeled as an energy storage device that could be charged and discharged.

Results from simulating GPS fix rates of 1 second, 30 seconds, and 15 minutes suggest SOC cannot be regulated using a fixed data collection rate, which could adversely effect the amount of data collected over time as well as the consistency by which GPS data is collected. Evaluation of accuracy, relative to an actual device, was performed to determine if the model was suitable for system development and testing. By performing simulation of operating conditions for an actual device, a comparison between modeled and calculated (based on measured battery voltage) SOC was used to evaluate model accuracy. Of the four devices evaluated, each over a period of 30 days, a maximum mean difference of 5% SOC was observed. In addition, none of the simulations had a deviation greater than 13% from actual data. Based on these findings, error was deemed within an acceptable range therefore the model may be used for system development and testing.



## Chapter 4

# System Development

### 4.1 Chapter Overview

Methodology for developing a control system to reduce variation in data collection rate, as well as maintaining a fully charged battery, is presented. PID control is presented as a method of adjusting a GPS receiver's data collection rate, to achieve energy equilibrium for a wildlife behavior monitor. Gain scheduling is shown to be used for switching between sets of PID gain values to produce a constant GPS fix interval. Methods for tuning two sets of PID gain values are shown and system performance based on available energy and disturbances is presented.

### 4.2 Controller Design

#### 4.2.1 Overview

Application of PID control to change GPS data collection, to achieve energy equilibrium for a wildlife behavior monitor, is presented. Error  $\epsilon(k)$  was generated at discrete time steps based on a difference in SOC between a set point  $SOC_{SetPoint}$  and modeled  $SOC(k)$  value. Based on the magnitude of  $\epsilon(k)$ , a gain scheduler selected one of two sets of gain values ( $K_P$ ,  $K_I$ , and  $K_D$ ) to be used by the controller to serve two purposes: (1) give the controller the ability to resist changing the GPS fix interval when the magnitude of the error function was within a threshold, and (2) adjust the GPS fix interval when the magnitude of the error function exceeded a set point threshold. Selected gain values  $K_P$ ,  $K_I$ , and  $K_D$ , as well as  $\epsilon(k)$  were taken as inputs to

the PID controller and an adjustment in GPS fix interval  $\Delta GFI$  (time, in seconds) was output as a control signal. Adjustments made to GPS fix interval were then applied to the GPS model and as a result, the rate at which energy was consumed also changed. Delay time of 5 minutes was used between subsequent controller calls to allow battery SOC to change in response to the adjustment in GPS fix interval. Figure 4.1 shows a diagram of the system that was described above.

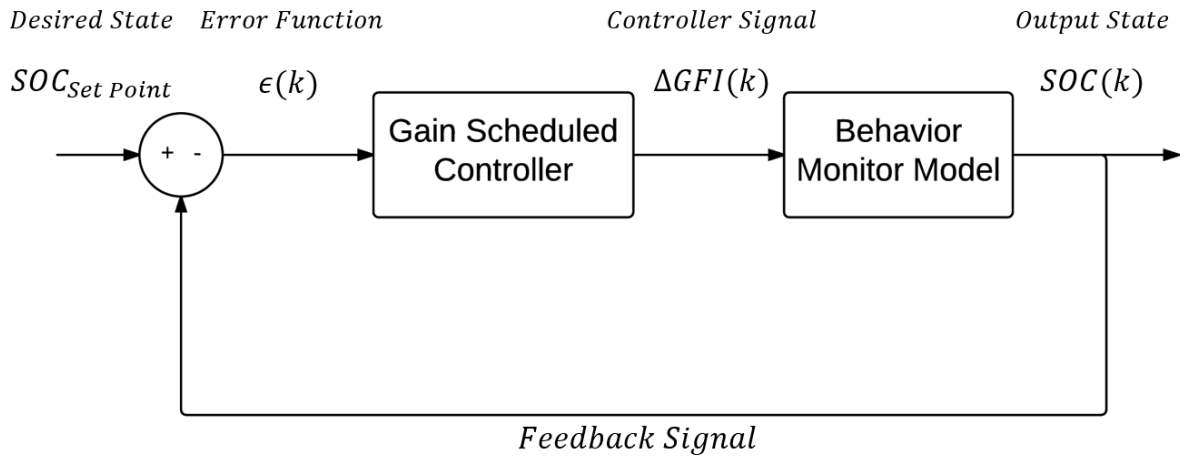


FIGURE 4.1: Diagram for an overview of system implementation.

#### 4.2.2 PID Control

PID control is presented as a method to develop a system for modifying the rate at which data was collected for a behavior monitor. Eq. 2.2 showed that an error signal  $\epsilon(k)$  was generated at discrete time intervals based on a system output state  $PV(k)$  and a set point  $S_P$ .  $\epsilon(k)$  was applied to energy consumption of a behavior monitor by assigning SOC as an output state  $SOC(k)$  and using a reference battery SOC as a set point  $SOC_{SetPoint}$ .

PID control, introduced in Eq. 2.3, was extended for GPS data collection by using  $\epsilon(k)$  from Eq. 2.2 as input to a PID controller and GPS fix interval was output.  $\epsilon(k)$  was used to generate a control signal  $\Delta GFI$  by the sum of three separate terms: proportional, integral, and derivative action. Proportional action was applied by multiplying a proportional gain constant  $K_P$  by the error  $\epsilon(k)$  at the current time step. The amount of change in the output signal could be increased by setting larger values of  $K_P$ . Integral action contributed to the output signal based on history of past error by adding error from each sampled time, scaled by  $K_I$  and the sampling period  $T$ . Derivative action produced a change in output signal based on the slope of error, or derivative, over time. The use of discrete sampling required a numerical approximation, therefore backward

difference was used between current  $\epsilon(k)$  and previous  $\epsilon(k-1)$  error divided by sampling period  $T$ . The degree of contribution of derivative action was weighted by a derivative gain value  $K_D$ , and could be tuned to account for future error. Refer to Figure 4.2 for a block diagram of PID control implemented for GPS data collection.

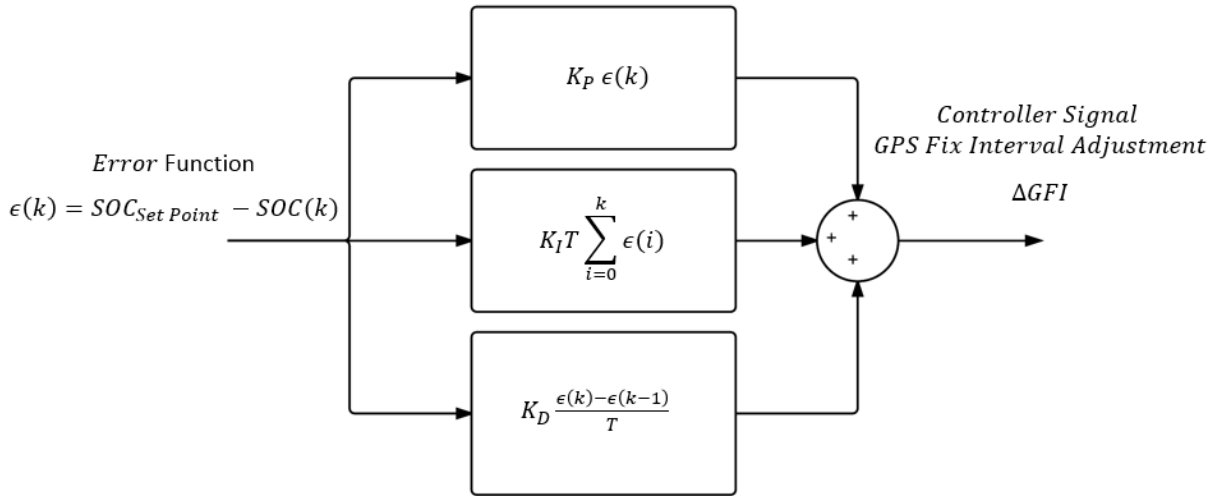


FIGURE 4.2: Block diagram of a PID controller.

### 4.2.3 Gain Scheduler

Two major requirements for system development were to maintain a fully charged battery and a consistent GPS fix interval for the behavior monitor. Due to variability of energy harvested from solar radiation, achieving equilibrium may be unlikely through a GPS spending energy, on data collection, at a constant rate. Two potential solutions for maintaining a fully charged battery and consistent GPS fix interval are as follows: (1) energy equilibrium was considered to be met for a range of values instead of a constant for battery SOC and as a result only minor ( $\pm 0-5$  seconds) modifications to data collection may need to be made. (2) Data collection, and consequently energy consumption could be modified for a GPS when the battery level drops.

PID controllers are typically used for linear systems and as a result may have unpredictable behavior for nonlinear systems. Nonlinearity of the PVC model, as well as maintaining both a consistent GFI and energy equilibrium, makes the use of a single PID controller unsuitable for achieving system requirements. For these reasons, gain scheduling, or assigning values based on different operation levels, was chosen by defining two sets of gain parameters ( $K_{P1}$ ,  $K_{I1}$ ,  $K_{D1}$  and  $K_{P2}$ ,  $K_{I2}$ ,  $K_{D2}$ ) that could be used by a PID controller. Figure 4.3 shows the block diagram

for selecting which set to use. Two operation levels were defined based on the magnitude of the error function from Eq. 2.2: (1) an error magnitude  $|\epsilon(k)| > 10\%$  considered to be at energy equilibrium, and (2)  $|\epsilon(k)| \geq 10\%$  was outside of equilibrium. Prior to a controller call, the gain scheduler evaluated  $|\epsilon(k)|$  and selected the corresponding set of gains to feed the controller.

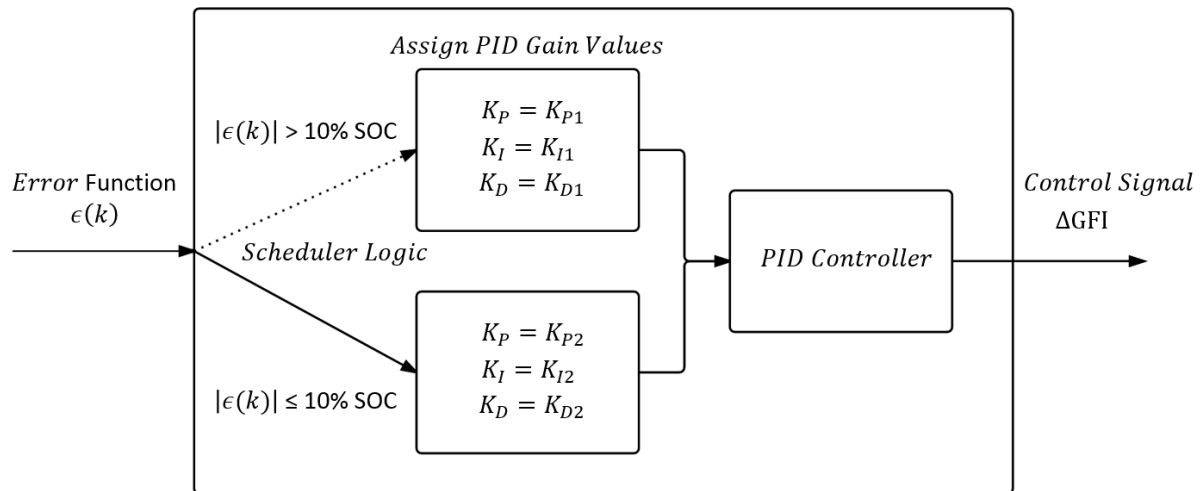


FIGURE 4.3: Block diagram for selecting PID gain values.

## 4.3 Controller Gain Tuning

Design methodology is presented for tuning a gain scheduled PID controller capable of maintaining a consistent GPS data collection rate based on SOC of a lithium ion battery. Selection of initial gain values is shown, where an energy consumption model was used to determine impact of gain adjustment on battery SOC and GPS data collection. Tuning is presented as a three stage process where each set of gains are tuned using a trial and error method. Results of gain tuning are presented and a comparison of performance is discussed for a scheduled versus a single set of PID gains.

### 4.3.1 Tuning Methodology Overview

Parameters of the energy consumption model and methodology used for tuning controller gains is presented. Tuning was performed in three stages: (1) Gains for a single controller were tuned for a constant value of a PVC model. (2) Single controller gains were adjusted for a variable PVC model. (3) Gain parameters for second controller, of a dual system, are tuned for a variable PVC model. Model parameters used for these three stages is presented below.

The first step for tuning a single set of PID gains was to reduce complexity of the energy consumption model by generating constant current for PVC model instead of current that varied based on a solar diurnal cycle. 10 mA was selected (maximum of 150 mA) because it represented a worst case scenario for the amount current being generated by the PVC. Simulations lasted for 24 hours, where the GPS model operated continuously, and energy consumption was sampled every second. Reference point for SOC was selected as 80% to keep the battery nearly full (GPS operation was disabled if SOC dropped below 30%). Finally, Battery capacity was reduced from 800 mA-hr to 100 mA-hr to increase sensitivity of battery SOC to changes in energy consumed and harvested.

The second step consisted of implementing a variable PVC model and making necessary adjustments to previously acquired gain values. The variable PVC model had a maximum current generation of  $10 \pm 2$  mA at 10:30 (midday), and a minimum value of 0 mA 6:00 am (sunrise) and 4:00 pm (sunset). The same PVC model was used for each day of a simulation to restrict causes of SOC, and GPS fix interval, change to modification of gain values. Because the PVC model was time variant, the GPS model was configured to only collect data during the day (between 6:00 am and 4:00 pm). Simulations were initially performed over two and three days to evaluate

transient response, but were extended to 30 days to measure consistency of GPS fix interval over time. The last step of tuning involved determining gains for a second controller of a dual controller system using the same model parameters as the second stage, with the exception of battery capacity being set to a commonly used value of 800 mA-hr.

### 4.3.2 Constant PVC model for a single controller

#### 4.3.2.1 Proportional Gain

The first step for tuning a single controller was to initialize  $K_{I1}$  and  $K_{D1}$  to zero. Battery SOC and GPS fix interval were then evaluated over 24 hours for  $K_{P1}$ . Plot (a) of Figure 4.4 shows that for  $K_{P1}$  values of 1, 0.7, and 0.3, SOC varied  $\pm 4\%$  every 3 hours,  $\pm 5\%$  every 4 hours, and  $\pm 7\%$  every 5 hours (of an 80% reference SOC) respectively. Plot (b) shows that GPS fix interval varies from 10 to 50 seconds every 3, 4, and 5 hours for  $K_{P1}$  values of 1, 0.7, and 0.3 respectively. A value of 0.3 was selected for  $K_{P1}$  because it produced the largest period of oscillation for the GPS fix interval and subsequently the slowest rate of change.

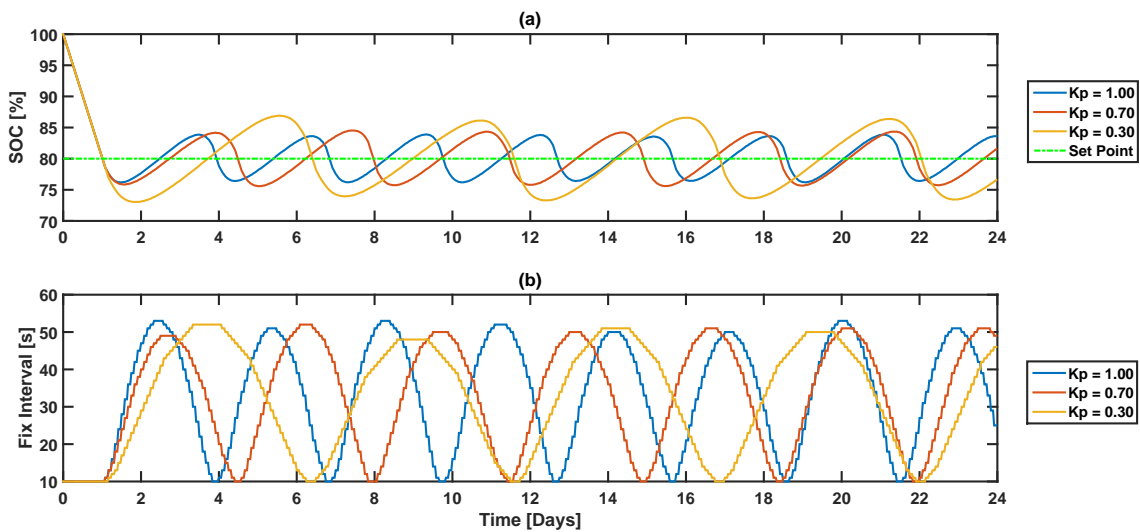


FIGURE 4.4: Plot (a) shows SOC with respect to time. Plot (b) shows the GPS fix interval.

#### 4.3.2.2 Derivative Gain

The second step for tuning a single controller was to leave the previously acquired  $K_{P1}$  (0.3) and  $K_{I1}$  (0) gains unaltered. Battery SOC and GPS fix interval were then evaluated for variable values of  $K_{D1}$ . Plot (a) of Figure 4.5 shows that for  $K_{D1}$  evaluated at 500, 1000, and 2000,

SOC had an overshoot of 3%, 2%, and 1% with times to reach an 80% reference point of 2, 4, and 5 hours respectively. Plot (b) shows a steady GPS fix interval of 25 seconds was reached in 6, 4, and 2 hours for  $K_{D1}$  values of 500, 1000, and 2000 respectively. A  $K_{D1}$  value of 500 was selected because it had a rise/steady state time of 2 hours for GPS fix interval.

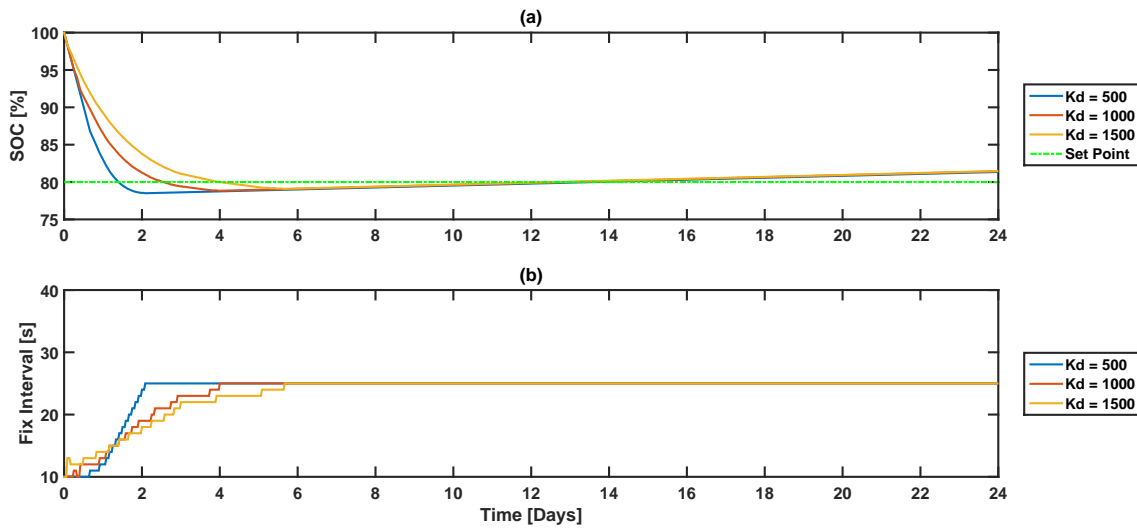


FIGURE 4.5: Plot (a) shows SOC with respect to time. Plot (b) shows the GPS fix interval.

### 4.3.2.3 Integral Gain

$K_{I1}$  was evaluated for battery SOC and GPS fix interval using  $K_{P1}$  of 0.3 and a  $K_{D1}$  of 2000. Simulations were performed with  $K_{I1}$  evaluated at 0, 0.001, and 0.01. Plot(a) of Figure 4.6 shows instability for each value except 0; for a  $K_{I1}$  of 0.001, SOC drops from 60% at 3 hours, 50% and 10 hours, and 40% at 24 hours. Plot(b) shows GPS fix interval alternating between the minimum value of 10 seconds and the maximum of 3600 seconds. Non-zero  $K_{I1}$  values for integral control caused undesirable oscillation for both SOC and data collection, therefore a value of 0 was used. Steady state error may be present without integral control, which is acceptable within a  $\pm 5\%$  threshold of a SOC set point.

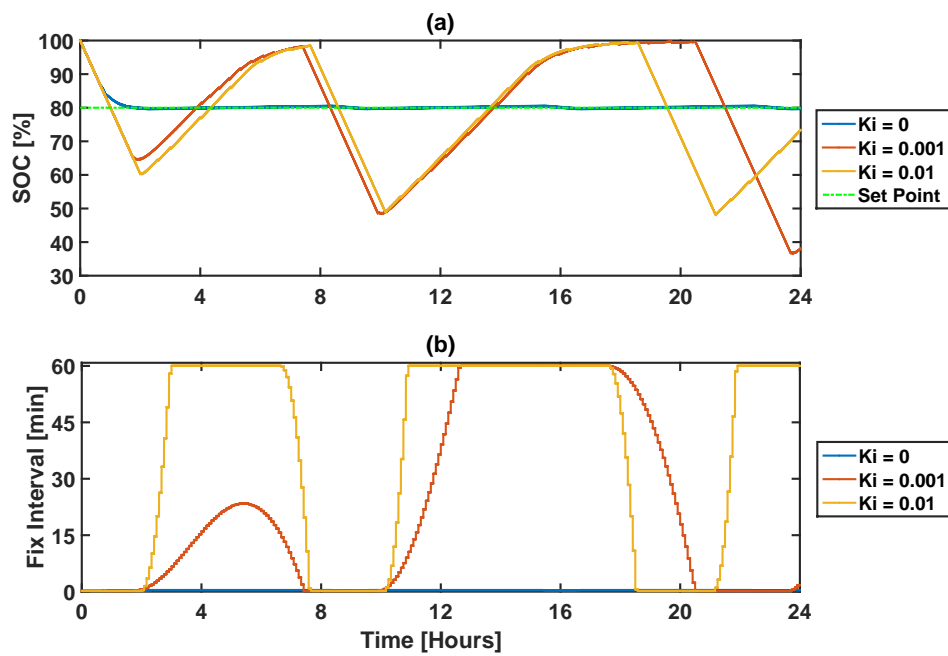


FIGURE 4.6: Plot (a) shows SOC with respect to time. Plot (b) shows the GPS fix interval.



#### 4.3.2.4 Result

Gain values for a single PID controller were tuned with a constant PVC current of 10 mA and a 100 mA-hr battery.  $K_{P1}$ ,  $K_{I1}$ , and  $K_{D1}$  were tuned one at a time and evaluated based on battery SOC and GPS fix interval.  $K_{P1}$  was assigned a value of 0.3,  $K_{D1}$  was assigned 2000, and  $K_{I1}$  was 0. Plot (a) of Figure 4.7 shows simulation results for SOC that started at 100% SOC and reached a reference SOC point of 80% in one hour. GPS fix interval reached a steady state of 25 seconds in 2 hours. SOC didn't reach a steady value due to the GPS fix interval being restricted to integer values where the value needed to achieve SOC steady state was between 24 and 25 seconds. Steady state is non-existent when PVC current is no longer constant, therefore inability to reach steady state for a simplified simulation has minimal impact.

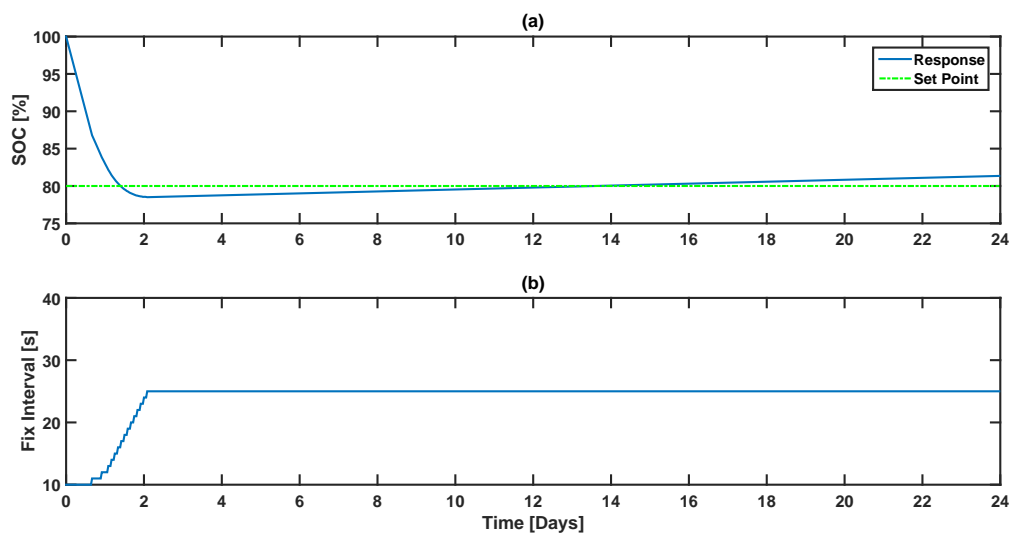


FIGURE 4.7: Plot (a) shows SOC with respect to time. Plot (b) shows the GPS fix interval.

### 4.3.3 Variable PVC model for a single controller

The second major step of tuning was performed for a PVC model that varied based on a solar diurnal cycle. Gain parameters for a single controller were re-tuned to account for changes in energy harvested throughout the day. GPS fixes were acquired during the day, which was dictated by the PVC model. Battery capacity remained at 100 mA-hr from the previous tuning step.

#### 4.3.3.1 Proportional Gain

Proportional gain was re-evaluated for battery SOC and GPS fix interval, given a change in the PVC model. Simulations were performed over two days to distinguish differences in response for different gains. Plot(a) of Figure 4.8 shows the PVC model which had a maximum current of  $10 \pm 2$  mA at midday. Plot (b) shows battery SOC for  $K_{P1}$  values of 0.3, 0.7, and 1. SOC varied between 75 and 85 percent for day 1 for all three gains. On day 2, SOC started and ended at 75% for  $K_{P1}$ 's of 0.7 and 1. For a  $K_{P1}$  of 0.3, SOC varied between 73 and 95 percent. Plot (c) shows that smaller values of  $K_{P1}$  resulted in less change in GPS fix interval.  $K_{P1}$  values of 0.3, 0.7 and 1 resulted in fix interval fluctuations from 10 seconds to 120, 170, and 260 seconds respectively. As a result, a  $K_{P1}$  value of 0.3 was used because it resulted in the lost magnitude of GPS fix interval fluctuation.

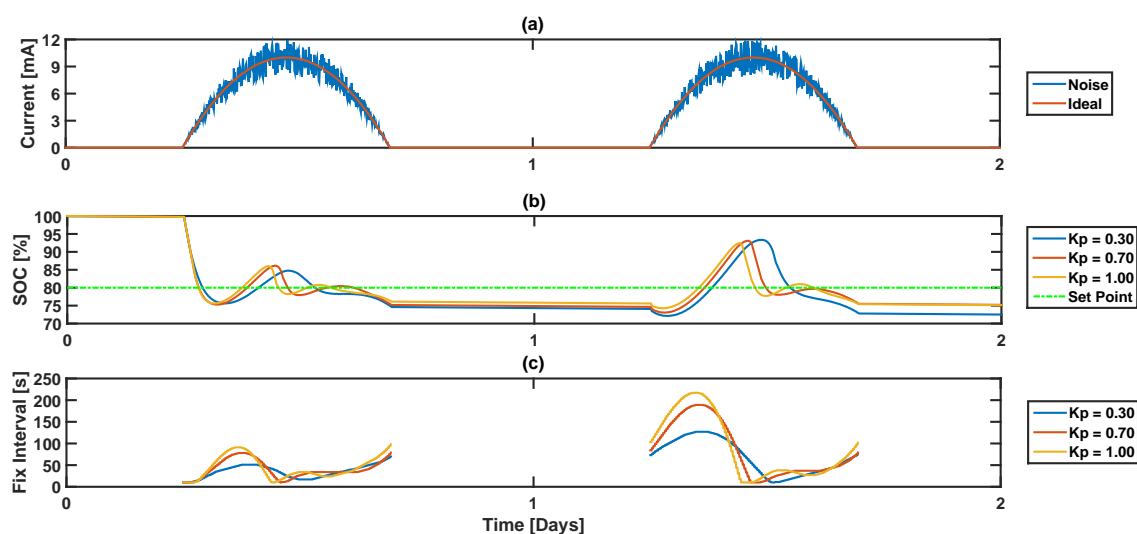


FIGURE 4.8: Plot (a) shows the variable PVC model. Plot(b) shows battery SOC. Plot(c) shows GPS fix interval.

### 4.3.3.2 Derivative Gain

Derivative gain for a single controller was re-evaluated for a variable PVC model. A simulation length of three days was needed to distinguish differences in SOC and GPS fix interval for  $K_{D1}$  values of 500, 1000, and 2000. Plot(a) of Figure 4.9 shows the PVC model which had a maximum current of  $10 \pm 2$  mA at midday. Plot(b) for  $K_{D1}$  values of 500, 1000, and 2000, SOC increased to 95, 90, 85 percent during the day and ended at 71, 75, and 75 percent respectively for day 3. Plot(c) showed GPS fix interval varied from 10 to 150, 120, and 110 for  $K_{D1}$  values of 500, 1000, and 2000 respectively. A value of 2000 was selected for  $K_{D1}$  because it resulted in the least variation in SOC and GPS fix interval for a variable PVC model.

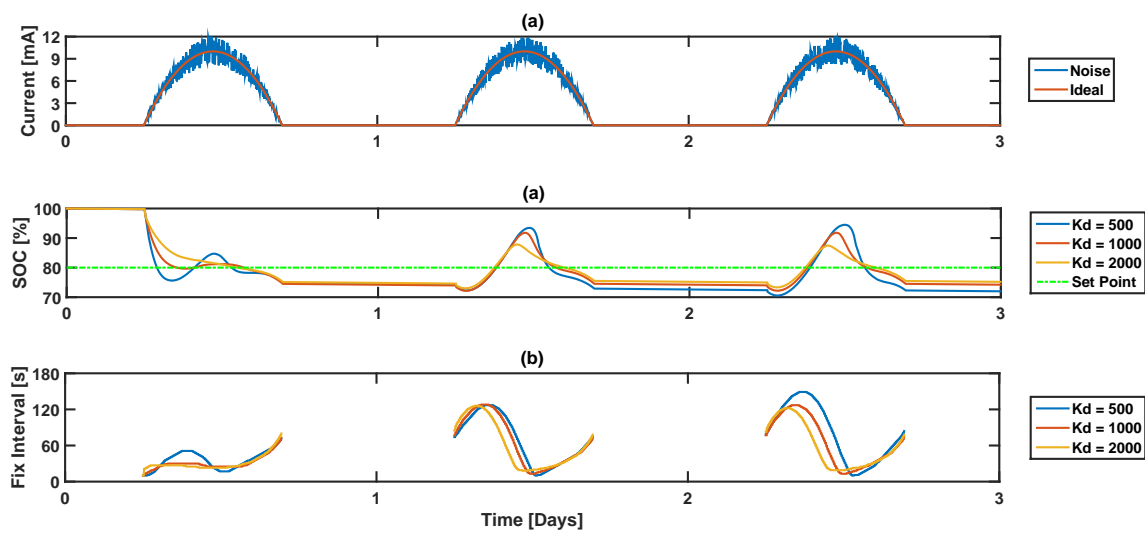


FIGURE 4.9: Plot (a) shows the variable PVC model. Plot(b) shows battery SOC. Plot(c) shows GPS fix interval.

### 4.3.3.3 Increased Battery capacity

A single controller with  $K_{P1}$ ,  $K_{D1}$ , and  $K_{I1}$  of 0.3, 2000, and 0 was evaluated based on GPS fix interval and SOC for a commonly used battery capacity of 800 mA-hr. Simulations were performed for 30 days, where Plot (a) of Figure 4.10 showed the PVC model for each day which had a maximum current of  $10 \pm 2$  mA at midday. Plot(b) showed SOC which varied periodically between 77 and 83 percent. Plot (c) showed variability of GPS fix interval between 10 and 60 seconds. After tuning a single controller, GPS fix interval was showed to vary based on a variable PVC model.

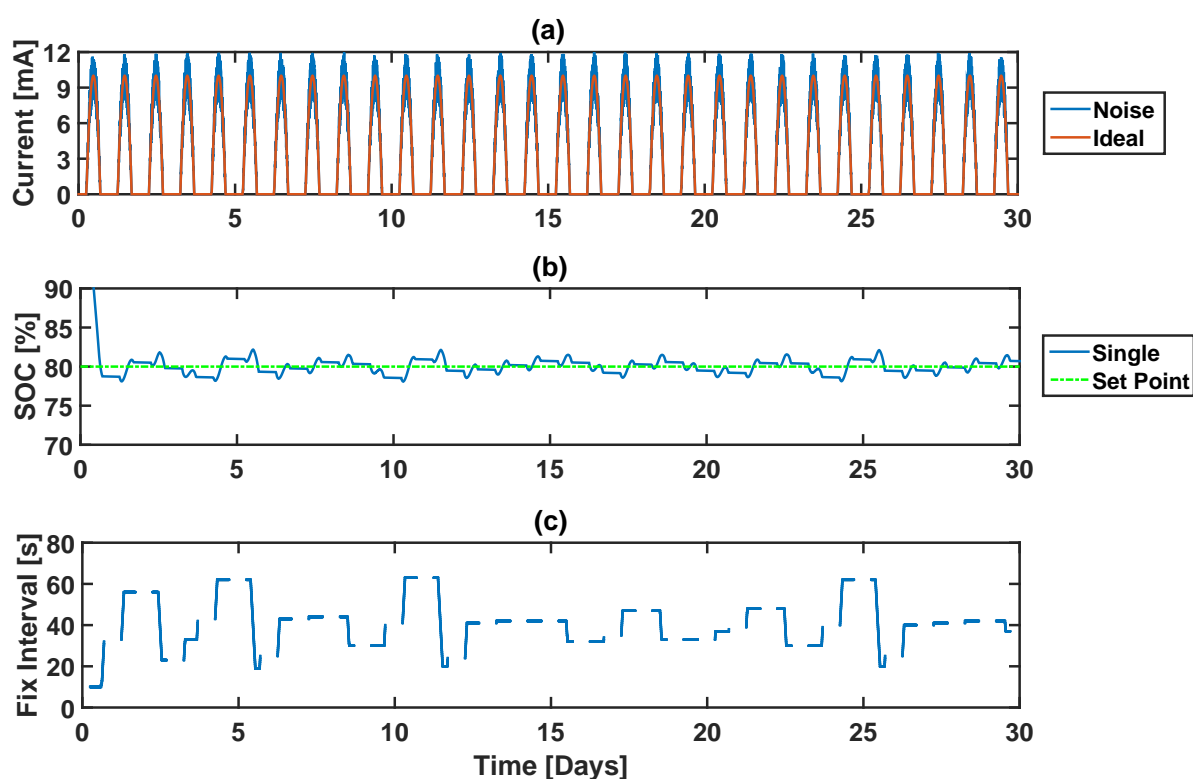


FIGURE 4.10: Plot (a) shows modeled PVC current. Plot(b) shows battery SOC. Plot(c) shows GPS fix interval.

#### 4.3.4 Variable PVC model for dual controllers

Tuning methodology is shown for two controller (gain scheduled) system. Determining which controller to use was done by the gain scheduler, introduced in Section 4.2.3. Tuning was performed by first setting gain values of the second controller  $K_{P2}$ ,  $K_{D2}$ , and  $K_{I2}$ , equal to the first  $K_{P1}$ ,  $K_{D1}$ , and  $K_{I1}$ . Simulations were performed for 30 days, battery capacity was 800 mA-hr, and the PVC model was variable.

##### 4.3.4.1 Proportional Gain

Three simulations were performed where  $K_{P2}$  was set to 0.3, 0.2, and 0.1 and SOC and GPS fix interval was evaluated. A  $K_{P2}$  value of 0.3 was used as a reference for comparison (Equivalent to  $K_{P1}$ ) to  $K_{P2}$  values of 0.2 and 0.1 Plot(a) of Figure 4.11 showed a variable PVC model where the maximum current generated was  $10 \pm 2$  mA at midday. Plot (b) showed SOC varied between 77 and 83 % for each proportional gain. Plot (c) showed for a  $K_{P2}$  value of 0.1, GPS fix interval reached a value of  $38 \pm 3$  percent in 6 days. For a  $K_{P2}$  value of 0.1, the fix interval reached a value of 38 percent in 25 days. A  $K_{P2}$  value of 0.1 was selected because it reached a consistent fix interval in 6 days.

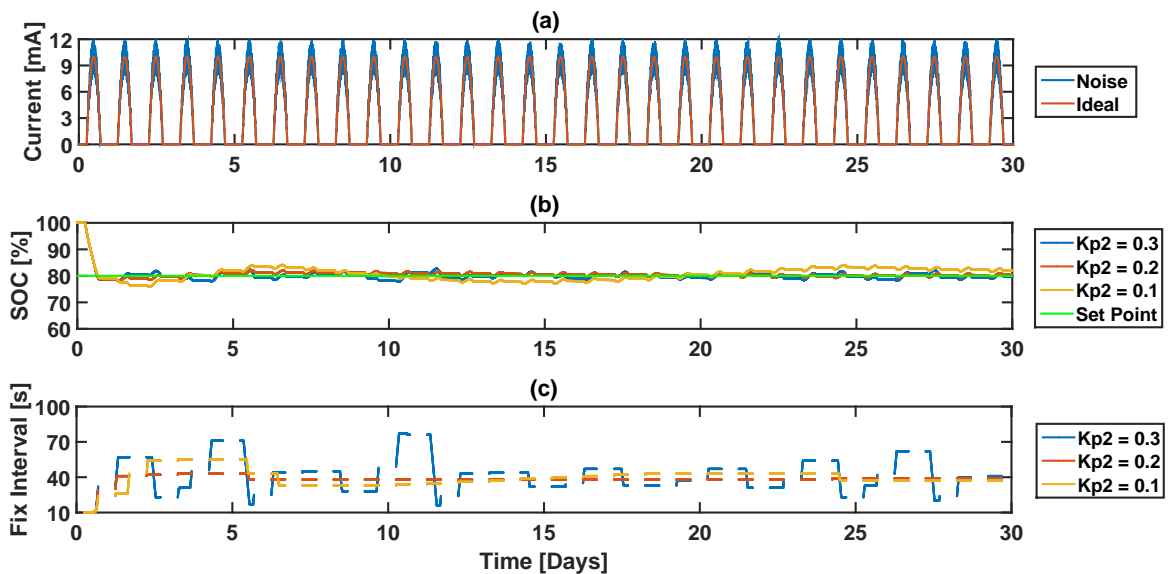


FIGURE 4.11: Plot (a) shows the variable PVC model. Plot(b) shows battery SOC. Plot(c) shows GPS fix interval.

#### 4.3.4.2 Derivative Gain

The next step in tuning the second controller was to select a value for  $K_{D2}$  where  $K_{P2}$  was previously selected as 0.1. Plot (c) of Figure 4.12 showed GPS fix interval for  $K_{D2}$  evaluated at 1000, 1500, and 2000. For  $K_{D2}$  set to 100, a GPS fix interval of  $38 \pm 2$  seconds was achieved after 28 days. During this transient period, GFI varied from 30 to 60 seconds. For  $K_{D2}$  set to 1500, a GPS fix interval of  $38 \pm 2$  seconds was achieved after 22 days. Ultimately a value of 2000 was used for  $K_{D2}$ , the same value as  $K_{D1}$ , because the time to reach a GPS fix interval was 6 days.

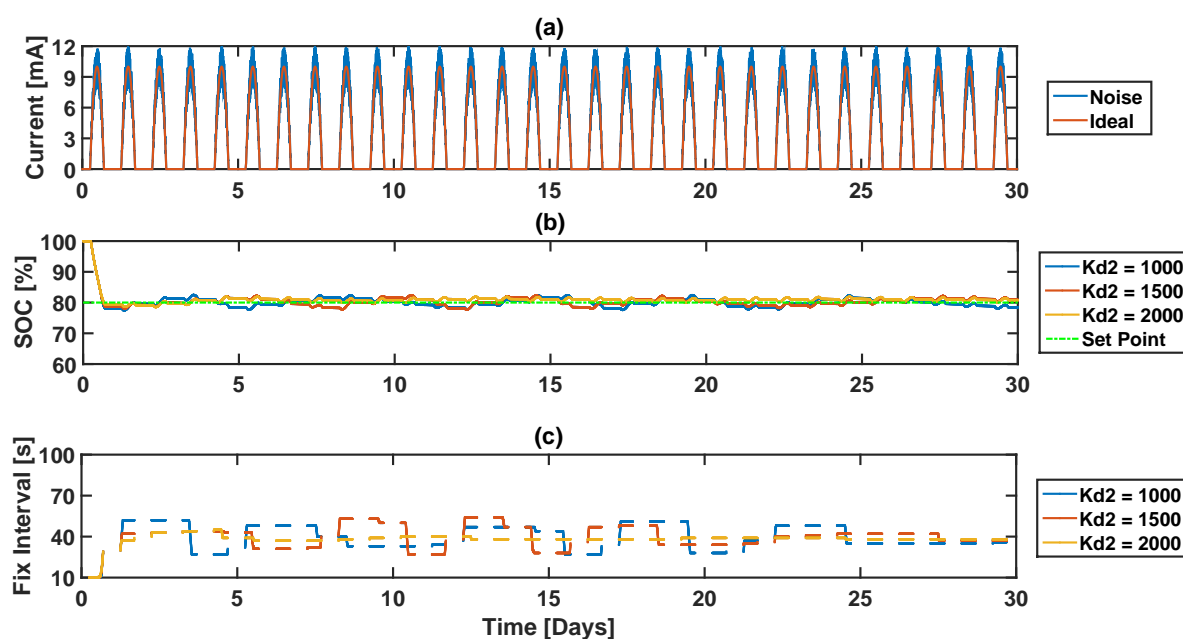


FIGURE 4.12: Plot (a) shows the variable PVC model. Plot(b) shows battery SOC. Plot(c) shows GPS fix interval.

### 4.3.5 Tuning Results

Tuning was performed for a system that used a discrete PID equation to modify GPS fix interval based on battery SOC. Gain values  $K_{P1}$ ,  $K_{I1}$ , and  $K_{D1}$  were tuned using a trial and error method where one gain parameter was tuned at a time given a constant PVC value of 10 mA.  $K_{P1}$  was assigned a value of 0.3,  $K_{D1}$  was assigned 500, and  $K_{I1}$  0 because SOC and GPS fix interval became unstable for non-zero gain values. Tuning was then performed using a variable PVC model and an increase in battery capacity from 100 mA-hr to 800 mA-hr.  $K_{P1}$ ,  $K_{I1}$ , and  $K_{D1}$  were also tuned using trial and error, where one gain parameter was tuned at a time.  $K_{P1}$  was assigned a value of 0.3 and  $K_{D1}$  was 2000. A second set of PID gains were added to account for fluctuation in GPS fix interval caused by a variable PVC model.  $K_{P2}$  was assigned a value of 0.1,  $K_{D2}$  was assigned 2000, and  $K_{I2}$  was 0.

Table 4.1 shows the results from tuning for a dual controller system where the controller used was determined by the gain scheduler. Two controllers were used, where one maintained a consistent GPS fix interval, and the other drove battery SOC to a reference point. Gain values denoted with a subscript of 1 refer to the controller used for  $\epsilon(k)$  greater than 10%, and a subscript of 2 represents gain values used that corresponding to  $\epsilon(k)$  less than 10%.

Gain Type	Set 1	Set 2
Proportional	$K_{P1} = 0.3$	$K_{P2} = 0.1$
Integral	$K_{I1} = 0$	$K_{I2} = 0$
Derivative	$K_{D1} = 2000$	$K_{D2} = 2000$

TABLE 4.1: Gain values for a dual controller system.

A comparison of SOC and GPS fix interval consistency was performed for gain scheduled (two gain set system) and a single controller. Simulations were performed over a period of 30 days and battery SOC and GPS fix interval were evaluated. Plot (a) of Figure 4.13 showed a variable PVC model where a maximum current of  $10 \pm 2$  mA occurred at each midday. For both controllers, Plot (b) showed that SOC reached the reference value in one day, and both remained within 4% of the SOC reference. For plot (c), GPS fix interval varied from 20 to 63 seconds over 30 days for a single controller. For the dual controller system, GPS fix interval settled at  $38 \pm 2$  seconds after 6.5 days.

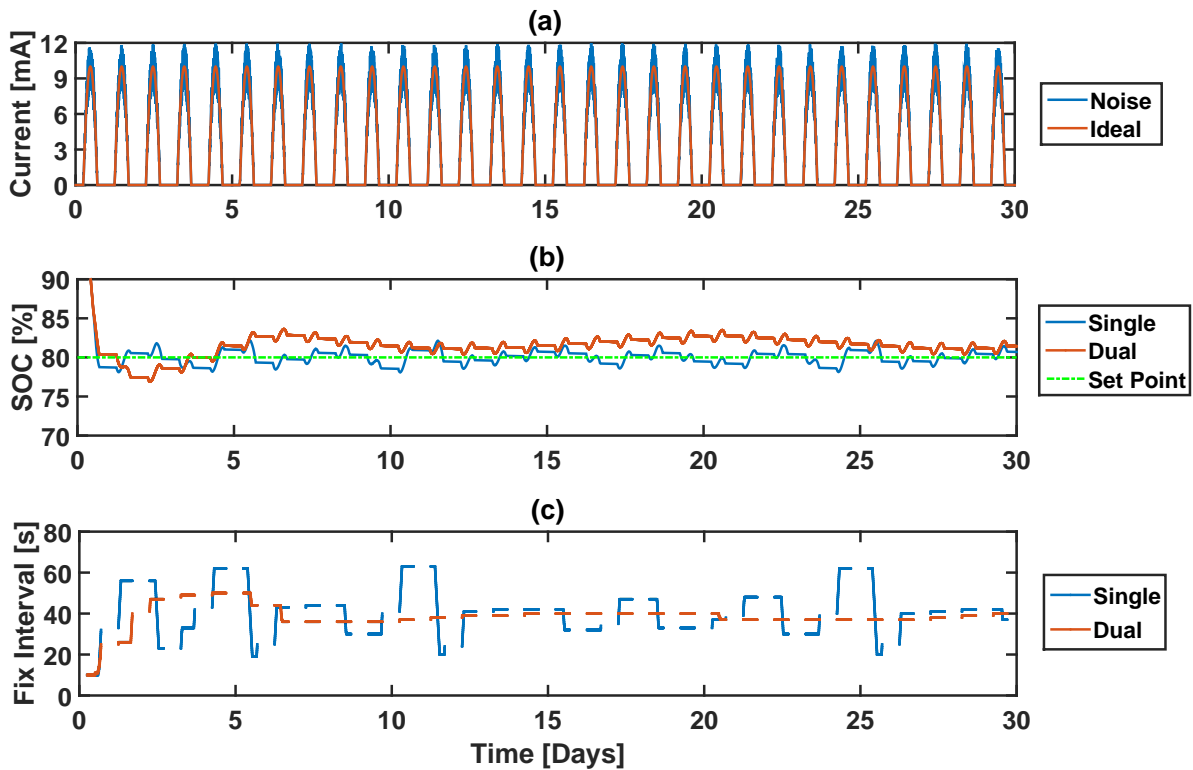


FIGURE 4.13: Plot (a) shows the variable PVC model. Plot (b) shows battery SOC. Plot (c) shows GPS fix interval.

## 4.4 Controller Design Summary

Gain scheduled, PID control was presented for limiting time for which the CTT-1100 tracking device's battery energy was unavailable and to maintain a consistent data collection rate. Battery SOC was a controlled parameter and an adjustment to GPS fix interval was a manipulated variable. Gain scheduling was used to switch between two controllers based on the difference between SOC and a reference point to produce a consistent GPS fix interval. Three steps for



tuning controller gains were presented: (1) A constant PVC model for 24 hours using a single controller, (2) a variable PVC model for 2 and 3 days using a single controller, and (3) a variable PVC model for a gain scheduled PID controller system. Model parameters for each step were outlined and a trial and error approach was used for determining proportional, integral, and derivative gains. Comparison of gain scheduled and standard controller system showed the use of gain scheduling outperformed a standard controller for maintaining a consistent GPS fix interval.

## 4.5 System Evaluation

Although solar radiation recurs in a pattern every 24 hours on Earth, factors such as weather changes, time of year, or PVC obstruction may cause changes in the amount of energy harvested each day. Evaluate is performed of a changeable data collection rate system's ability to adjust and maintain a GPS fix interval to maintain battery SOC for a change in available harvested energy. Design methodology is presented for applying both an increase and decrease in energy generated by a PVC then a comparison is shown for system performance against a constant GPS fix interval.

### 4.5.1 Decrease in Available Harvested Energy

Evaluation of system performance, with respect to battery SOC and GPS fix interval, was done by scaling energy harvested one half after several days of operation. Simulation was performed over a period of 40 days where the PVC model varied in proportion to a solar diurnal cycle. For the first twenty days, a maximum current of  $10 \pm 2$  mA occurred at midday then scaled down by 50% for the next twenty days. The decrease occurred on day 20 to ensure SOC and GPS fix interval were at a steady state before performing the experiment. The PVC model can be seen in Plot (a) of Figure 4.14. Plot (b) shows the system controlled SOC started at 100% and reached the reference point of 80% SOC in 10 hours. On day twenty, SOC dropped from 82% to a minimum of 75.8% over three days. For a constant GPS fix interval of 30 seconds, SOC decreased by 2% each day until day 20, then decreased by 6% until day 25 at which point the GPS was shut off. Plot (c) showed that a GPS fix interval started at 10 seconds to overcome the initial difference in SOC and settled at 38 seconds with a 16% overshoot that lasted 5 days. For day 20, the GPS fix interval increased to 110 seconds for 7 days then settled at  $78 \pm 2$  seconds for the remainder of the simulation.

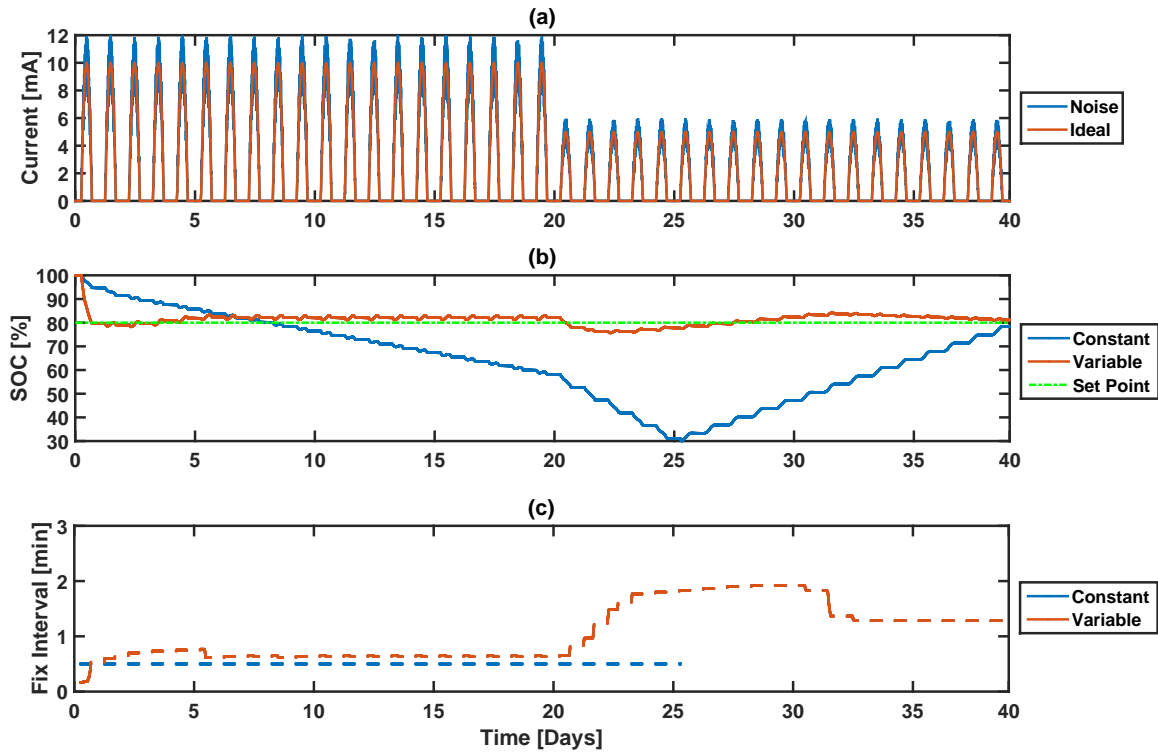


FIGURE 4.14: Plot (a) shows the modeled PVC current. Plot (b) shows SOC. Plot (c) shows GPS fix interval.

#### 4.5.2 Increase in Available Harvested Energy

SOC and GPS fix interval were evaluated for an increase in energy harvested over a period of 40 days, where the PVC model had a current of  $5 \pm 2$  mA for the first 20 days, then  $10 \pm 2$  mA for the last 20 days as seen in Plot (a) of Figure 4.15. Plot (b) showed the system controlled SOC started at 100% and reached the reference point of 80% SOC in 9 hours after starting the simulation. On day twenty, SOC increased from 83.3% to a maximum value of 83.51% over a day. SOC for a constant GPS fix interval decreased 5% each day until day 13 when the GPS was forced off because the 30% threshold was reached. The battery was charged to 80% at day 24 then began decreasing again. Plot (c) shows the GPS fix interval started at 10 seconds to overcome the initial difference in SOC, due to a fully battery, and settled at  $78 \pm 2$  seconds after being at  $120 \pm 3$  seconds until day 13. Due to an increase in energy harvested on day 20, GPS fix interval decreased to 35 seconds in two days to overcome the rise in SOC. The fix interval then settled at  $38 \pm 3$  seconds for the remainder of the simulation.

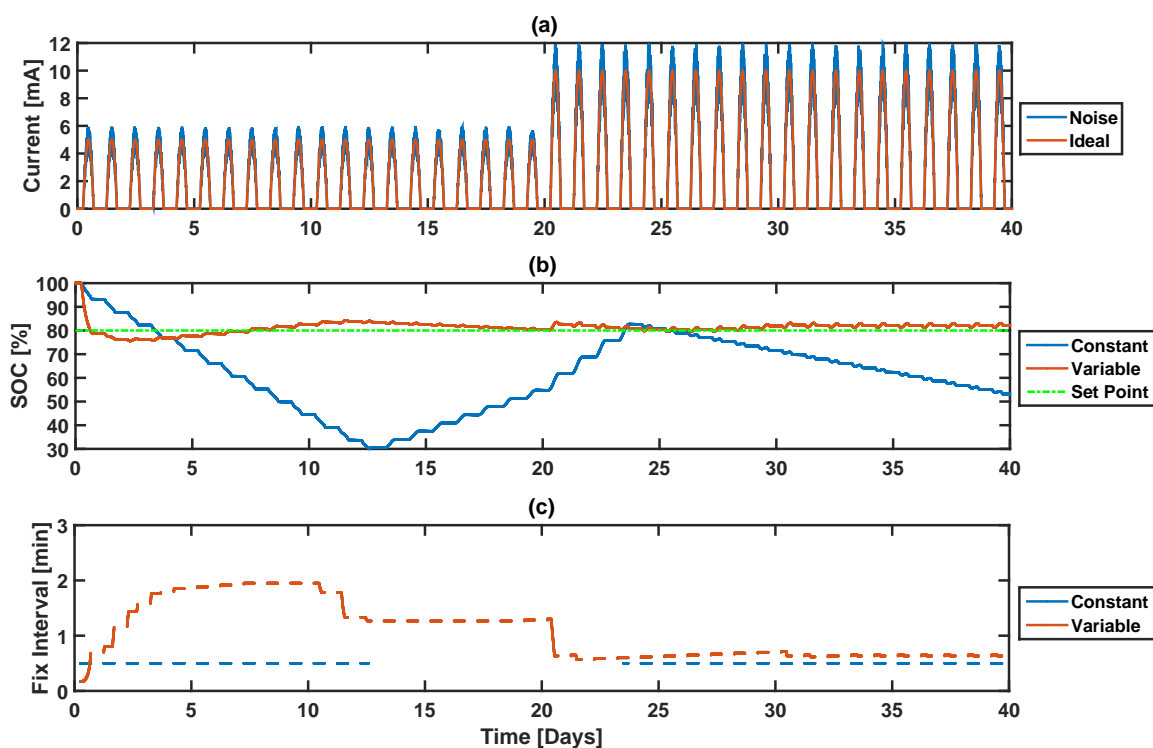


FIGURE 4.15: Plot (a) shows the modeled PVC current for each day. Plot(b) shows SOC with respect to time. Plot(c) shows the GPS fix interval over time.

### 4.5.3 Summary

The impact of modifying energy harvested on SOC and GPS fix interval was introduced and compared to a constant GPS fix interval. For both an increase and decrease in harvested energy, SOC remained within a  $\pm 10\%$  SOC threshold when controlled by the changeable data collection rate system. For a constant GPS fix interval, SOC decreased until the GPS was forced off. The system was shown to adjust GPS fix interval to a constant level ( $\pm 3$  seconds) for changes in available energy. The next section introduces system performance when exposed to disturbance.

## 4.6 Disturbance Handling

Many factors such as cellular data transfer, battery capacity variation due to rapid temperature change, or inability for a GPS receiver to acquire data may effect the systems ability to adjust and maintain a consistent data collection rate as well as battery SOC. In order to understand the impact external disturbances on system performance, two disturbance cases are presented that may cause uncertain controller behavior. (1) A drop in battery SOC due to cellular data communication and (2) an unexpected increase in time necessary for a GPS to collect a fix.

### 4.6.1 Cellular Disturbance

The first disturbance involved instantaneously reducing battery SOC and evaluating the time necessary for the GPS fix interval to recover to its value before the disturbance. Drop in battery SOC could result from cellular data transfer due to energy required therefore an evaluation was performed over 40 days where the PVC model had variable current output proportional to a solar diurnal cycle. For each day, a maximum current generation of  $10 \pm 2$  mA occurred at midday. The disturbance was performed on day 11 of the simulation to allow SOC and GPS fix interval to be at a steady state before the evaluation. For day 11, SOC was instantaneously decreased and the time necessary to recover to the initial GPS fix interval was evaluated. Based on Figure 4.16, six simulations were performed where the magnitude of SOC drop was varied from 5% to 30% by increments of 5%. For the smallest drop of 5%, the GPS fix interval returned to its original value in 5 days, a 10% SOC dropped took 13 days, 15% SOC drop took 15 days, 20% took 17 days, 25% took 19 days, and 30% took 20 days. The amount of recovery time increased on average by 2 days for each 5% drop in SOC.

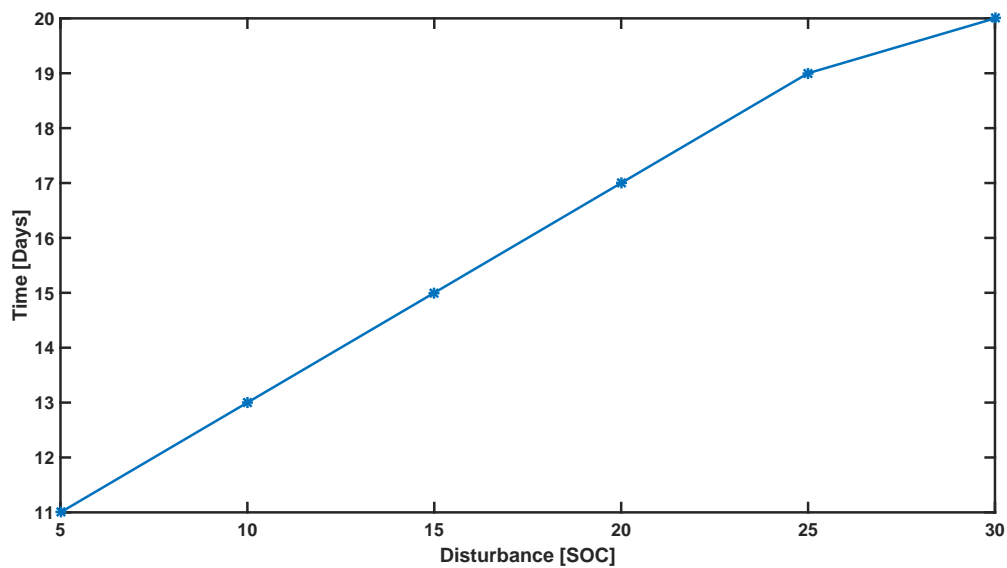


FIGURE 4.16: Disturbance in battery SOC.

### 4.6.2 GPS Disturbance

The second disturbance case increased the length of time necessary to acquire a GPS fix which resulted in additional energy consumption due to the GPS receiver being active for an extended period. Evaluation was performed over 40 days where the PVC model had variable current output proportional to a solar diurnal cycle. For each day, a maximum current generation of  $10 \pm 2$  mA occurred at midday. Disturbance increased the time to acquire a GPS fix by a factor of 10, and lasted from one to five days. Figure 4.17 shows recovery time for the GPS fix interval was 10 days when the disturbance lasted 1 day. For two days, recovery time increased to 12.5 days, 14.2 days for a three day disturbance, 15.5 days for four days, and 15.9 days for a five day disturbance. GPS disturbance would be typical in densely covered areas such as forests or cities with large buildings where GPS signal is weak.

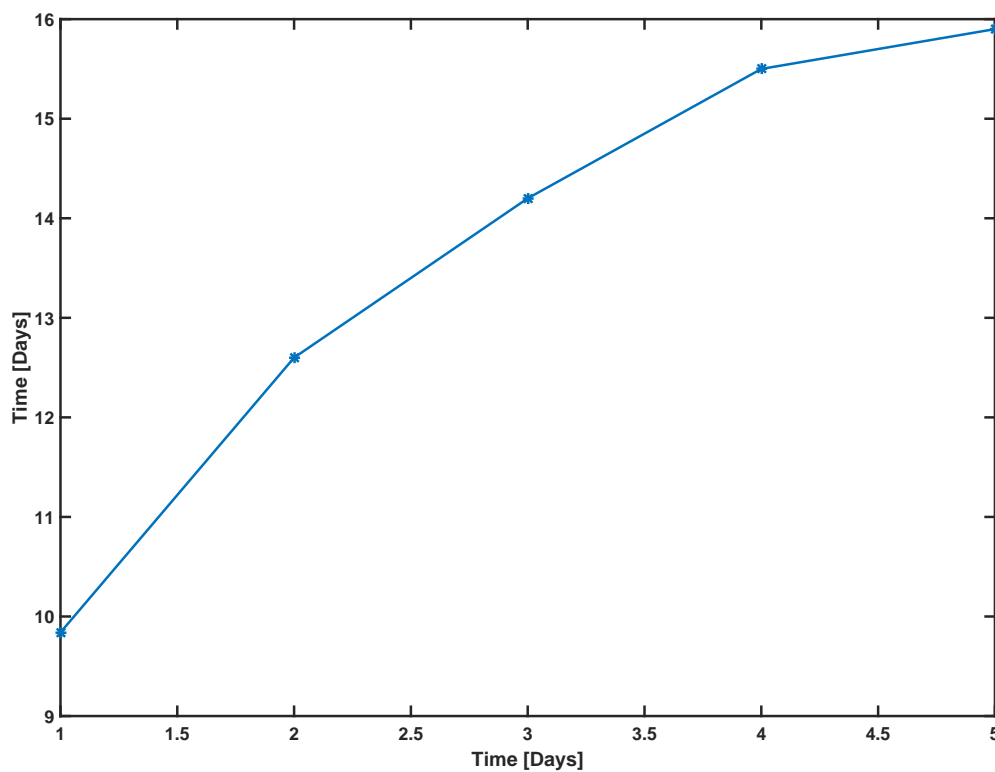


FIGURE 4.17: Disturbance in time necessary to acquire a GPS fix.

### 4.6.3 Disturbance Summary

Controller performance was evaluated based on two cases of disturbance. The first case involved dropping SOC instantaneously by values ranging from 5% to 30% to simulate large current consumption associated with cellular communication. Recovery time varied from 11 to 20 days based on the magnitude of drop in battery charge. The second case increased the time necessary to acquire a GPS fix as a result of poor GPS signal strength. Recovery time ranged from 10 to 16 days. While the system was able to recognize and attempt to correct disturbances in SOC, it required extensive amounts of time to return to a normal GPS fix interval.

## 4.7 Chapter Conclusion

Gain scheduling was introduced for a PID controller system to maintain a consistent GPS fix interval, and limit the amount of time battery charge was unavailable. GPS fix interval, for a behavior monitor, was manipulated based on an error signal between a measured and

reference battery SOC. The magnitude of control error was then evaluated by a gain scheduler to determine which set of gains was to be used. One controller was designed to maintain a consistent GPS fix interval, and the other for changing the interval to maintain SOC at a reference point. Three steps for tuning a gain scheduled controller system were presented: (1) A constant PVC model for 24 hours using a single controller, (2) a variable PVC model for 2 and 3 days using a single controller, and (3) a variable PVC model for a gain scheduled controller system. Model parameters for each step were outlined and a trial and error approach was used for determining proportional, integral, and derivative gains. Based on tuning results, it was clear that non-zero integral gain values had a negative impact on controller performance therefore it was not used. Comparison was performed for a gain scheduled and standard controller system that showed that using gain scheduling outperformed a standard PID controller for maintaining a consistent GPS fix interval.

The impact of modifying energy harvested on SOC and GPS fix interval was introduced and compared to a constant GPS fix interval. For both an increase and decrease in harvested energy, SOC remained within a  $\pm 10\%$  SOC threshold when system controlled. For a constant GPS fix interval, SOC decreased until the GPS was forced off. The system was shown to adjust GPS fix interval to a constant level ( $\pm 3$  seconds) for changes in available energy. System performance was also evaluated based on two cases of disturbance. The first case involved dropping SOC instantaneously by values ranging from 5% to 30%. Acute battery SOC disturbance may have occurred during cellular communication. Recovery time varied linearly from 11 to 20 days based on the magnitude of drop in battery charge. The second case involved an increase in time necessary to acquire a GPS fix. Recovery time ranged from 10 to 16 days. While the system was able to recognize and attempt to correct disturbances in SOC, it required extensive amounts of time to return to a normal GPS fix interval.



## Chapter 5

# Conclusion and Future Work

### 5.1 Thesis Summary

#### 5.1.1 Problem Overview

Behavior monitors typically collect data, and consequently spend energy, at fixed intervals. Dependent on the rate at which energy is harvested, impact on battery charge can be classified in three ways. (1) Battery charge may become depleted over time because energy consumed by data collection is greater than energy harvested therefore data collection must stop until enough energy has been harvested to continue. (2) Energy may be harvested faster than it can be consumed which can result in a battery becoming fully charged and unharvested energy could be applied to collecting more data. (3) Energy spent on data collection may be balanced with energy harvested and therefore battery depletion doesn't occur and data collection continues. The third case, where energy harvested is balanced with energy consumed, is preferred because data can continually be collected at a maximum level that doesn't result in battery depletion. Adjusting a data collection rate, proportional to changes in battery charge, was achieved through analysis and design of a data collection compensation system such that data obtained was maximized without sacrificing battery energy sustainability.

#### 5.1.2 Energy Consumption Model

Energy consumption of the CTT-1100 wildlife behavior monitor was modeled for development and evaluation of a changeable data collection rate system, where GPS, PVC, microcontroller,

and real time clock cumulatively dictated available battery energy. Software architecture was analyzed for developing behaviors for each component of the model. The GPS receiver's energy consumption was modeled based on a rule set that related time to acquire a measurement to the data collection rate. PVC current generated from solar energy harvesting was modeled proportionally to a solar diurnal cycle. Random noise was also incorporated and the magnitude of energy harvested could be scaled differently for each day. Operation of a microcontroller was represented by an active and sleep current and a real time clock applied a constant level of current at all times. Finally, a lithium ion battery was modeled as an energy storage device that could be charged and discharged.

Results from simulating GPS fix rates of 1 second, 30 seconds, and 15 minutes suggest SOC cannot be regulated using a fixed data collection rate, which could adversely effect the amount of data collected over time as well as the consistency by which GPS data is collected. Evaluation of accuracy, relative to an actual device, was performed to determine if the model was suitable for system development and testing. By performing simulation of operating conditions for an actual device, a comparison between modeled and calculated (based on measured battery voltage) SOC was used to evaluate model accuracy. Of the four devices evaluated, each over a period of 30 days, a maximum mean difference of 5% SOC was observed. In addition, none of the simulations had a deviation greater than 13% from actual data. Based on these findings, error was deemed within an acceptable range therefore the model may be used for system development and testing.

### **5.1.3 System Design and Evaluation**

Gain scheduling was used for a PID controller system to maintain a consistent GPS fix interval, and limit the amount of time battery charge was unavailable. GPS fix interval, for a behavior monitor, was manipulated based on an error signal between a measured and reference battery SOC. The magnitude of control error was then evaluated by a gain scheduler to determine which set of gains was to be used. One controller was designed to maintained a consistent GPS fix interval, and the other for changing the interval to maintain SOC at a reference point. Three steps for tuning a gain scheduled controller system were presented: (1) A constant PVC model for 24 hours using a single controller, (2) a variable PVC model for 2 and 3 days using a single controller, and (3) a variable PVC model for a gain scheduled controller system. Model parameters for each step were outlined and a trial and error approach was used for determining proportional, integral, and derivative gains. Based on tuning results, it was clear that non-zero

integral gain values had a negative impact on controller performance therefore it was not used. Comparison was performed for a gain scheduled and standard controller system that showed that using gain scheduling outperformed a standard PID controller for maintaining a consistent GPS fix interval.

The impact of modifying energy harvested on SOC and GPS fix interval was introduced and compared to a constant GPS fix interval. For both an increase and decrease in harvested energy, SOC remained within a  $\pm 10\%$  SOC threshold when system controlled. For a constant GPS fix interval, SOC decreased until the GPS was forced off. The system was shown to adjust GPS fix interval to a constant level ( $\pm 3$  seconds) for changes in available energy. System performance was also evaluated based on two cases of disturbance. The first case involved dropping SOC instantaneously by values ranging from 5% to 30%. Acute battery SOC disturbance may have occurred during cellular communication. Recovery time varied linearly from 11 to 20 days based on the magnitude of drop in battery charge. The second case involved an increase in time necessary to acquire a GPS fix. Recovery time ranged from 10 to 16 days. While the system was able to recognize and attempt to correct disturbances in SOC, it required extensive amounts of time to return to a normal GPS fix interval.

The designed system will have impacts such as: (1) Behavior monitors would automatically configure their GPS fix interval based on available energy. (2) A consistent data collection rate, at a given level of harvested energy, could be easily implemented into statistical models for behavior. (3) By maintaining a fully charged battery, operational life would be extended. Overall this system could improve functionality, usability, and life expectancy of a wildlife behavior monitor.

## **5.2 Discussion**

### **5.2.1 Scientific Contribution**

Contribution to the field of engineering was twofold: (1) A software model of a behavior monitor was developed as a tool for simulating energy consumption of sensors and components, energy harvested, and battery charge over time. Modules for additional sensors or components could be added based on future need. (2) Methods for dynamic data collection were developed with the ability to maintain consistent data collection and limit battery charge depletion. Changeable

GPS data collection was developed based on research from multiple fields including modeling, control, and energy management of wireless sensor nodes.

### **5.2.2 Future Work**

Future work may be performed in the following areas:

1. Performance needs to be evaluated on an actual behavior monitor.
2. Empirical data for battery level and solar energy needs to be gathered at a 30 second sampling period to more accurately validate the model.
3. System performance could be tested using actual data for solar energy.
4. Additional controllers could be added, through gain scheduling, to reduce time necessary for GPS fix interval to recover from disturbance.

## Appendix A

# Simulation Script

Functions for performing simulation of energy consumption for a wildlife behavior monitor are presented. SimulateExecute.m takes simulation setup properties as input and returns useful data such as battery charge, solar current generation, and GPS data collection rate. Initialization of various hardware components found on a behavior monitor such as GPS, PVC, MCU, and a battery are included. The user has the ability to enable an variable data collection rate system that has a two stage gain schedule. Values for both sets of gains can be passed as input. PVC current data can be generated based on three modes: Simple mode provides a flat current value for the entirety of the simulation. Normal mode generates a parabolic profile for current that is proportional to a daily diurnal cycle. Real mode allows the user to specify a text file that contains current data. Below is the source code for SimulateExecute.m.

---

```

1 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2 %
3 %           SimulationExecute.m
4 %
5 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
6 function [SOC_,SOC_avg_,REALSOC_, FR_, PV_, PVavg_,GPS_] = ...
    SimulationExecute(days_,PV_MODE,RealDay,bool-gps-simple,Initial_GFI,solar_,...
    Battery_Properties,USE_SOCFilter,GS,Kp,Kd,Ki,Kp2,Kd2,Ki2,Disturb,DisturbMag)
7
8 watchdog = true;           % Causes GPS to turn off for SOC < 30%
9 filename = 'EmpiricalData.csv'; % Filename for Data Validation
10 filestart = 2;           % First Line of Empirical Data
11 filestop = 1093;        % Last Line of Empirical Data

```

```

12
13 days = days_; % Length (days) of Simulation
14 t_span = 0:24*3600-1; % Number of Seconds in 1 day
15 Time.Absolute = 1; % Initialize index for time variable
16
17 % Controller Properties
18 controller_call = 300; % Length of time between Controller Calls [s]
19 threshold = 10; % Threshold for Switching Controller Gains
20 isFirstCall = true; % Initializes Controller Variables
21 Set_Point = Battery_Properties(3); % Reference State of Charge for Controller
22 State = ones(1,days*24*3600); % Holds Gain Schedule state of Controller
23
24
25 % Battery Properties
26 Batt_max = Battery_Properties(1); % Capacity of Battery [mA-s]
27 Batt_Level = Battery_Properties(2); % Battery Charge [mA-s]
28 Battery = BatteryClass(days,Batt_Level,Batt_max,3600); %% Instantiate Battery
29
30
31
32 #####
33 %           Moving Average SOC           #
34 #####
35 wind = 0.5*3600; % Window Size for Moving Average
36 inst_soc = zeros(2,days*24*3600); % Preallocate Array
37 ind = 1; % First element of Moving Average SOC
38 point_soc(1) = Battery.GetBatterySOC(); % Get first element
39 point_soc(2,ind) = 0; % Corresponding time
40 ind = ind + 1; % Iterate to second index
41
42 #####
43 %           Configure PV           #
44 #####
45 DayTime = 6*3600; % Start of Day for Solar Charging 6:00 am
46 NightTime = 16.67*3600; % End of Day for Solar Charging 4:00 pm
47 Max_PV = 15; % Maximum amount of current for PVC [mA]
48 PVMaxLen = length(solar_); % Number of elements in Current Array
49
50 % Make sure there are enough Current values in the array for each day
51 assert(PVMaxLen >= days, 'PVMAX must have at least %d inputs\n',days)
52

```

```

53 #####
54 %             PV Mode Config             #
55 #####
56 %             Value      |      Description
57 %-----
58 %             0         |      Simple Mode
59 %             1         |      Normal Mode
60 %             2         |      Read Mode
61 %-----
62
63 % Simple Mode: Constant charge current for 24 hours
64 % Normal Mode: Day/Night cycle. Parabolic charge profile
65 % Real Mode: Day/Night cycle and charge specified by real data
66
67 % Instantiate PhotoVoltaic Cell Object
68 PV = PVClass(PV_MODE,days,DayTime,NightTime,0,solar_(1),Max-PV);
69
70 #####
71 %             GPS Config                 #
72 #####
73 if PV_MODE == 2           % If Real Mode
74     FixRate = 15*60;      % GPS Fix rate used by Actual Device
75 else
76     FixRate = Initial_GFI; % Initialize GPS fix rate
77 end
78
79 FixLength = 60;          % Time to First Fix           [s]
80 Min_FR = 10;            % Minimum fix rate GPS can have [s]
81
82 isSOCDisturb = false;   % Determines if SOC disturbance is occurring
83 isFLDisturb = false;   % Determines if GPS disturbance is occurring
84 % Instantiate GPS Object
85 GPS = GPSClass(bool_gps_simple,days,FixRate,FixLength,DayTime,NightTime);
86
87 #####
88 %             Conservative Enable        #
89 #####
90 User_Select_Conservative = false;      % Set true to increase the minimum GPS ...
          fix interval
91 if User_Select_Conservative == true    % Increases minimum fix rate

```

```

92     Min_FR = GPS.CalculateMinimumFixRate(solar_(1), SOC_F, Set_Point, Batt_max, ...
      DayTime, NightTime);
93 end
94
95 % Instantiate MCU/RTC object
96 MCU = MCUClass();
97
98
99 %% Time Loop for 'd' Days
100 Time = zeros(1,days*24*3600);           % Holds Absolute Time
101 avg_soc = zeros(1,days*24*3600);        % Holds AVG SOC
102 SumofEnergy = zeros(1,length(t_span)); % Holds Energy consumed at each time step
103 REAL_SOC = [];                          % Initialize Real SOC array (Model ...
      Validation)
104 daylen = length(t_span);                % Number of seconds in a day
105
106 #####
107 %           Main Time Loop [Days]           #
108 #####
109 for d = 1:days
110
111     GPS.ClearFixes();    % Clear number of fixes acquired
112     controller_call = 0; % Controller Call Variable clear
113
114     if PVMaxLen == 1     % Gets index for PVC profile to use
115         pv_pos = 1;
116     else
117         pv_pos = d;
118     end
119
120     if PV_MODE == 1     % If Normal Mode, Generate PVC profile
121         [PV.SunCycle, PV.AvgSunCycle] = PV.GenerateSunCycle(DayTime, NightTime, ...
      solar_(pv_pos));
122     elseif PV_MODE == 2 % Acquire Actual PVC data from File
123         clear PV.SunCycle DayTime
124         if days == 1
125             [PV.SunCycle, SOC_rtemp, DayTime] = AcquireActualPVCDData(RealDay, ...
      filename, filestart, filestop);
126             lenpv = length(SOC_rtemp);
127             temp_ = isnan(SOC_rtemp);
128

```



```

129         for k = 1:lenpv
130             if temp_(k) ≠ 1
131                 start_ = k;
132                 break;
133             end
134         end
135         Batt_Level = Batt_max*SOC_rtemp(start_)/100;
136         Battery.Battery_Level = Batt_Level*3600;
137     else
138         [PV.SunCycle, SOC_rtemp, DayTime] = AcquireActualPVCDData(d, filename, 2, 1487)...
;
139     end
140
141     REAL_SOC = cat(2, REAL_SOC, SOC_rtemp);
142     PV.Sun_Start = DayTime;
143     GPS.Start_Time = DayTime;
144     end
145     %% Time Loop for 24 Hours
146     tic;
147
148     #####
149     %           Inner Time Loop [Seconds]           #
150     #####
151     for cnt = 1:daylen
152
153         t = t_span(cnt);
154         Time(Time_Absolute) = Time_Absolute;
155
156         if GPS.isActive()
157             SumofEnergy(cnt) = PV.GetUsage(t) - GPS.GetUsage(t) - MCU.MCU_Active;
158         else
159             SumofEnergy(cnt) = PV.GetUsage(t) - GPS.GetUsage(t) - MCU.MCU_Sleep;
160         end
161
162
163         if Disturb(1) == 1 % GPS Disturbance
164             if (d ≥ Disturb(2) && d ≤ Disturb(3))
165                 isFLDisturb = true;
166             else
167                 isFLDisturb = false;
168             end

```

```

169     elseif Disturb(1) == 2 % Battery Disturbance
170         if (t == 12*3600) && (d == Disturb(2))
171             isSOCDisturb = true;
172         else
173             isSOCDisturb = false;
174         end
175     end
176
177     Battery.UpdateBattery(Time_Absolute, SumofEnergy(cnt), isSOCDisturb, DisturbMag) ...
178     ;
179     GPS.UpdateGPS(Time_Absolute);
180     PV.UpdatePV(Time_Absolute,t);
181
182     #####
183     %           SOC Moving Average           #
184     #####
185     if USE_SOCFilter % Implements moving Average SOC if enabled
186
187         inst_soc(Time_Absolute) = Battery.GetBatterySOC();
188
189         if Time_Absolute ≤ wind
190             start = 1;
191         else
192             start = Time_Absolute-wind;
193         end
194
195         temp_soc = inst_soc(start:Time_Absolute);
196         avg_soc(Time_Absolute) = mean(temp_soc);
197         SOC_FW = avg_soc(Time_Absolute);
198
199     end
200
201     #####
202     %           Controller Call           #
203     #####
204
205     if PV.isDayTime(t) && PV_MODE ≠ 2 % Day Time and not real data use
206         if controller_call == 0
207
208             if USE_SOCFilter

```

```

209         [FixRate, State(Time.Absolute)] = GetFixRate(isFirstCall, FixRate, ...
Call_Period, SOC_FW, Set_Point, Min_FR, threshold, GS, Kp, Kd, Ki, Kp2, Kd2, Ki2);
210         isFirstCall = false;
211     else
212         [FixRate, State(Time.Absolute)] = GetFixRate(isFirstCall, FixRate, ...
Call_Period, Battery.GetBatterySOC(), Set_Point, Min_FR, threshold, GS, Kp, Kd, Ki, ...
Kp2, Kd2, Ki2);
213         isFirstCall = false;
214     end
215
216     GPS.SetFixInterval(FixRate, isFLDisturb);
217     controller_call = Call_Period-1;
218
219     else
220         controller_call = controller_call - 1;
221     end
222
223     GPS.UpdateFixInterval(Time.Absolute, true);
224
225     elseif PV_MODE == 2
226         GPS.SetFixInterval(FixRate, isFLDisturb);
227         GPS.UpdateFixInterval(Time.Absolute, true);
228     else
229         GPS.UpdateFixInterval(Time.Absolute, false);
230     end
231
232     % Increment the Absolute Time Index
233     Time.Absolute = Time.Absolute + 1;
234
235     end % End Inner Loop
236 end % End Outer Loop
237
238 % Return Values from Function to User
239 SOC_.. = (Battery.Battery_Array/Battery.Battery_Capacity)*100;
240 SOC_avg_.. = avg_soc;
241 FR_.. = GPS.Fix_Interval_Array(1, :);
242 PV_.. = PV.PV_Array;
243 PVavg_.. = PV.PV_AVG_Array;
244 GPS_.. = GPS.GPS_Usage_Array;
245 REALSOC_.. = REALSOC;
246

```

```
247 end % End Function
```

---

## Appendix B

# Hardware Classes

Appendix B introduces source code for the GPS, PVC, and battery classes. Each class was instantiated in `SimulateExecute.m` and used to generate the net energy applied to the battery.

### B.1 Global Positioning System

---

```

1 classdef GPSClass < PowerClass & handle
2     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
3     %                               Properties                               %
4     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5     properties
6         % Current Properties
7         GPS_OFF          = 0;
8         GPS_Off_Sleep   = 0.015;      % Sleeping, Off [mA]
9         GPS_Fix_Sleep   = 0.441;      % Sleeping, Has Fix [mA]
10        GPS_Fix_Active  = 27.5;       % Active, Has Fix [mA]
11
12        % Active Times
13        Start_Time;
14        End_Time;
15
16        % Fix Times
17        TimetoFirstFix   = 60;
18        Current_Fix_Length;
19        Normal_Mode_Fix_Length = 60;    % Time to acquire a fix from OFF [s]

```

```

20     Fast_Mode_Fix_Length = 7;           % Time to acquire a fix (Fast Mode) [...
    s]
21     Continuous_Default_Fix_Length = 10; % Time to acquire a fix (Continuous ...
    Mode) [s]
22
23     % Data Storage
24     GPS_Usage;           % Current GPS State [mA]
25     GPS_Usage_Array;    % Stores GPS Usage data over time [mA]
26     GPS_Consumption;    % The amount of energy consumed by the ...
    GPS module
27
28     % Fixes
29     Fix_Length;         % Time GPS takes to get fix (Varies ...
    based on Collection Mode)
30     Number_Fixes;      % Number of Fixes Acquired [integer]
31     Fix_Interval;      % Time Between Fixes [s]
32     Fix_Interval_Array; % Stores the Fix Interval over time [s]
33     Simple_Mode;
34
35     Continuous_Mode_cnt;
36     Operate_cnt;
37     Operate_state;
38 end
39
40 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
41 %                               Methods                               %
42 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
43 methods
44
45 % GPS Class Constructor
46 function obj = GPSClass(Simple_Mode, Num_Days, Fix_Interval, Fix_Length, Start, Stop...
    )
47
48     obj.Simple_Mode = Simple_Mode;
49
50     obj.Fix_Interval = Fix_Interval;
51     obj.Fix_Length = Fix_Length;
52     obj.SetFixInterval(Fix_Interval, false);
53
54     obj.GPS_Consumption = 0;
55

```

```

56     % Set Default Values
57     obj.NumberFixes = 0;
58     obj.GPS_Usage = obj.GPS_Fix_Active;
59     obj.GPS_Usage_Array = zeros(1,Num_Days*24*3600);
60     obj.Fix_Interval_Array = zeros(2,Num_Days*24*3600);
61
62     if obj.SimpleMode
63         obj.Start_Time = 0;
64         obj.End_Time = 24*3600;
65     else
66         obj.Start_Time = Start;
67         obj.End_Time = Stop;
68     end
69 end % End Constructor
70
71
72 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
73 %             Getter/Setters             %
74 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
75
76 function out = GetFixes(obj)
77     out = obj.NumberFixes;
78 end
79
80 function ClearFixes(obj)
81     obj.NumberFixes = 0;
82 end
83
84 function out = GetFixInterval(obj)
85     out = obj.Fix_Interval;
86 end
87
88 function SetFixInterval(obj,Time,disturb)
89     obj.Fix_Interval = round(Time); % Set the Fix Rate to nearest integer ...
90     value
91     % Set the fix length
92     if Time >= 15*60
93         obj.Current_Fix_Length = obj.Normal_Mode_Fix_Length; % Set the ...
94     else
95         time to collect a fix

```

```

95         obj.Current_Fix_Length = obj.Fast_Mode_Fix_Length;
96     end
97
98     if disturb
99         obj.Current_Fix_Length = obj.Current_Fix_Length*10;
100    end
101 end
102
103 function out = GetFixLength(obj)
104     out = obj.Fix_Length;
105 end
106
107 function SetFixLength(obj,Time)
108     obj.Fix_Length = Time;
109 end
110
111 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
112 %           End Getter/Setters           %
113 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
114
115 function out = GetUsage(obj,Time)
116
117     if Time < obj.Start_Time || obj.Start_Time < 0
118         out = obj.Inactive();
119     elseif (Time ≥ obj.Start_Time) && (Time < obj.End_Time)
120         if Time ≤ (obj.Start_Time + obj.TimetoFirstFix + 1)
121             obj.Startup(Time);
122             out = obj.GPS_Usage;
123         else
124             if obj.Fix_Interval > 10
125                 obj.Operate(Time);
126                 out = obj.GPS_Usage;
127             else
128                 out = obj.ContinuousMode(Time);
129             end
130         end
131     else
132         out = obj.Inactive();
133     end
134
135     obj.GPS_Consumption = obj.GPS_Consumption + out;

```



```
136
137     if Time == 24*3600-1
138         obj.GPS_Consumption = 0;
139     end
140 end
141
142 function out = isActive(obj)
143     if obj.GPS_Usage == obj.GPS_Fix_Active
144         out = true;
145     else
146         out = false;
147     end
148 end
149
150 function Startup(obj, Time)
151     if Time ≤ obj.TimeToFirstFix + obj.Start_Time
152         obj.GPS_Usage = obj.GPS_Fix_Active;
153     else
154         obj.NumberFixes = obj.NumberFixes + 1;
155         obj.GPS_Usage = obj.GPS_Fix_Sleep;
156     end
157 end
158
159 function out = ContinuousMode(obj, Time)
160
161     if isempty(obj.ContinuousMode_cnt) || Time == obj.TimeToFirstFix + 1
162         obj.ContinuousMode_cnt = obj.Fix.Interval;
163     end
164
165     obj.GPS_Usage = obj.GPS_Fix_Active;
166     out = obj.GPS_Usage;
167     obj.ContinuousMode_cnt = obj.ContinuousMode_cnt - 1;
168
169     if obj.ContinuousMode_cnt == 0
170         obj.ContinuousMode_cnt = obj.Fix.Interval;
171         obj.NumberFixes = obj.NumberFixes + 1;
172     end
173 end
174
175 function Operate(obj, Time)
176     % Operate_state
```

```

177     % true = active
178     % false = sleep
179
180     if isempty(obj.Operate_state) || Time == obj.TimetoFirstFix + 1
181         obj.Operate_state = false;
182         obj.Operate_cnt = obj.Fix_Interval - obj.Current_Fix_Length-1;
183     end
184
185     if obj.Operate_state == false
186         if obj.Operate_cnt > 0
187             obj.GPS_Usage = obj.GPS_Fix_Sleep; %GPS_Off_Sleep;
188             obj.Operate_cnt = obj.Operate_cnt-1;
189         else
190             obj.GPS_Usage = obj.GPS_Fix_Active;
191             obj.Operate_cnt = obj.Current_Fix_Length;
192             obj.Operate_state = true;
193         end
194     else
195         obj.GPS_Usage = obj.GPS_Fix_Active;
196         obj.Operate_cnt = obj.Operate_cnt - 1;
197
198         if obj.Operate_cnt ≤ 0
199             obj.NumberFixes = obj.NumberFixes + 1;
200             obj.Operate_cnt = obj.Fix_Interval - obj.Current_Fix_Length-1;
201             obj.Operate_state = false;
202         end
203     end
204 end
205
206 function out = Inactive(obj)
207     obj.GPS_Usage = obj.GPS_OFF;
208     out = obj.GPS_Usage;
209 end
210
211 % Update GPS Array
212 function UpdateGPS(obj,time)
213     obj.GPS_Usage_Array(time) = obj.GPS_Usage;
214 end
215
216 function out = CalculateMinimumFixRate(PV,SOC_in,SP,Batt_cap,DayTime,...
NightTime)

```

```

217
218     aprox_gps_cost = 30*15;
219     aprox_pv_avg = (NightTime - DayTime)*2/3*PV;
220     diff_SP = abs((SP - SOC_in)/100*Batt_cap*3600);
221
222     % Balance of energy Eavail = Ein + Estored - Eout
223     Eavail = aprox_pv_avg + diff.SP;
224
225     fixes = Eavail/(aprox_gps_cost);
226
227     out = round((NightTime - DayTime)/fixes);
228
229     if out < 10 % The fix rate cannot go below 10 seconds
230         out = 10;
231     end
232
233 end
234
235 % Update Fix Interval Array
236 function UpdateFixInterval(obj,time,bool)
237     if bool == true % GPS is enabled
238         obj.Fix_Interval_Array(1,time) = obj.Fix_Interval;
239         obj.Fix_Interval_Array(2,time) = time;
240     else % GPS disabled, max fix interval
241         obj.Fix_Interval_Array(1,time) = NaN;
242         obj.Fix_Interval_Array(2,time) = time;
243     end
244 end
245
246 end % End Methods
247 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
248 %                               End Methods                               %
249 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
250
251 end % End Classdef

```

---

## B.2 Photo Voltaic Cell

---



```

38      %                                     Methods                                     %
39      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
40      methods
41
42      % Constructor
43      function obj = PVClass(PV_MODE, Num_Days, DayTime, NightTime,PV_Level,...
SolarMax,PVMax)
44          obj.PV_MODE = PV_MODE;
45          obj.mA = 0;
46
47          if obj.PV_MODE == 0                % SIMPLE MODE
48              obj.Sun_Start = 0;
49              obj.Sun_Stop = 24*3600;
50              obj.Sun_Duration = 24*3600;
51              obj.SolarMax = SolarMax;
52          elseif obj.PV_MODE == 1          % NORMAL MODE
53              obj.Sun_Start = DayTime;
54              obj.Sun_Stop = NightTime;
55              obj.Sun_Duration = NightTime - DayTime;
56              obj.SolarMax = SolarMax;
57              obj.PVMax = PVMax;
58              [obj.SunCycle, obj.AvgSunCycle] = obj.GenerateSunCycle(obj.Sun_Start,...
obj.Sun_Stop,SolarMax);
59              obj.SunCycleLength = length(obj.AvgSunCycle);
60          else                            % REAL MODE
61              obj.Sun_Start = DayTime;
62              obj.Sun_Stop = NightTime;
63              obj.Sun_Duration = NightTime - DayTime;
64          end
65
66          obj.PV_Array = zeros(1,Num_Days*24*3600);
67          obj.PV_AVG_Array = zeros(1,Num_Days*24*3600);
68          obj.PV_Level = PV_Level;
69          obj.PV_Cumulative = 0;
70          obj.PV_Daily_Cumulative = 0;
71      end %End Constructor
72
73      function [Act, Avg] = GenerateSunCycle(obj,DayTime,NightTime,Solar)
74          PV_Min_ = (1-0.2)*Solar;
75          PV_Max_ = (1+0.2)*Solar;
76

```

```
77     min = generate_parabola(DayTime,NightTime,PV_Min.);
78     max = generate_parabola(DayTime,NightTime,PV_Max.);
79     Solar_Ideal = generate_parabola(DayTime,NightTime,Solar);
80
81     Act = zeros(1,length(Solar_Ideal));
82
83     len = length(Solar_Ideal);
84     for i = 1:len
85         Act(i) = min(i) + (max(i)-min(i)).*rand(1,1);
86         if Solar_Ideal(i) > obj.PVMax
87             Solar_Ideal(i) = obj.PVMax;
88         end
89     end
90     Avg = Solar_Ideal;
91 end
92
93 % Get Current PV Level
94 function out = Get_PV(obj)
95     out = obj.PV_Level;
96 end
97
98 % Set Current PV Level
99 function Set_PV(obj,val)
100     obj.PV_Level = val;
101 end
102
103 % Set Max Value for solar intensity
104 function Set_PVMax(obj,val)
105     obj.SolarMax = val;
106 end
107
108 function out = Get_PVCumulative(obj)
109     out = obj.PV_Cumulative;
110 end
111
112 function Clear_PV_Cumulative(obj)
113     obj.PV_Daily_Cumulative = obj.PV_Daily_Cumulative + obj.PV_Cumulative;
114     obj.PV_Cumulative = 0;
115 end
116
117 function out = Get_PV_Daily_Cumulative(obj)
```

```

118     out = obj.PV_Daily_Cumulative;
119 end
120
121 function Clear_PV_Daily_Cumulative(obj)
122     obj.PV_Daily_Cumulative = 0;
123 end
124
125 % Update PV Array and PV Cumulative
126 function UpdatePV(obj,t_absolute, t_relative)
127
128     obj.PV_Cumulative = obj.PV_Cumulative + obj.PV_Level;
129     obj.PV_Array(t_absolute) = obj.PV_Level;
130
131     if obj.PV_MODE == 0 || obj.PV_MODE == 2
132         return;
133     end
134
135     if obj.isDayTime(t_relative)
136         if isempty(obj.ASCcnt) || obj.ASCcnt > obj.SunCycleLength
137             obj.ASCcnt = 1;
138         end
139
140         obj.PV_AVG_Array(t_absolute) = obj.AvgSunCycle(obj.ASCcnt);
141         obj.ASCcnt = obj.ASCcnt + 1;
142     else
143         obj.PV_AVG_Array(t_absolute) = 0;
144     end
145 end
146
147 function out = isDayTime(obj,t)
148
149     if obj.PV_MODE == 0 % Simple Mode, Always day time
150         out = true;
151     else % Otherwise, compare with start and stop times
152         if t < obj.Sun_Start
153             out = false;
154         elseif t > obj.Sun_Stop
155             out = false;
156         else
157             out = true;
158         end

```

```

159     end
160 end
161
162 function out = GetUsage(obj,t)
163
164     if obj.PV_MODE == 0                               % Simple Mode, Always on
165         obj.PV_Level = obj.SolarMax;
166     elseif obj.PV_MODE == 2                           % Real Mode, Data comes from ...
167         File
168         obj.PV_Level = obj.SunCycle(t+1);
169     else                                               % Normal Mode, Data comes ...
170         from GenerateSunCycle()
171         if t < obj.Sun.Start
172             obj.PV_Level = 0;
173         elseif t ≥ obj.Sun.Start && t < obj.Sun.Stop-1
174             index = t - obj.Sun.Start+1;           % Get index of solar curve
175             obj.PV_Level = obj.SunCycle(index);
176         else
177             obj.PV_Level = 0;
178         end
179     end
180 end % End GetUsage()
181
182 end % End Methods
183 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
184 %                               End Methods                               %
185 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
186 end % End Classdef

```

---

## B.3 Battery

```

1
2 classdef BatteryClass < handle
3
4     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
5     %                               Properties

```



```

6      %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7      properties
8          Battery_Capacity;    % Maximum Battery Charge [mA-s]
9          Battery_Level;      % Current Battery Level [mA-s]
10         Battery_Array;      % Array that Holds Battery Level with respect to Time
11         Battery_Average;    % Current Average Battery level over 24 hours. [mA-s]
12         Battery_Average_Array;
13         Average_Window;
14
15         Battery_Watchdog_MinLevel;
16         Battery_Watchdog_MaxLevel;
17
18     end
19     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
20     %                               End Properties
21     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
22
23
24     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
25     %                               Methods
26     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
27     methods
28         % Battery Class Constructor
29         function obj = BatteryClass(Num_Days,Batt_Level, Batt_Cap,Window)
30             obj.Average_Window = Window;
31             obj.Battery_Capacity = Batt_Cap*3600;           % Convert...
32             from [mA-h] to [mA-s]
33             obj.Battery_Level = Batt_Level*3600;          % Convert...
34             from [mA-h] to [mA-s]
35             obj.Battery_Average = obj.Battery_Capacity/3600; % [mA-hr]
36             obj.Init_WatchDog(30,90);                     % Set SOC...
37             ranges for WatchDog
38             obj.Battery_Array = zeros(1,Num_Days*24*3600);
39         end
40
41         % Calculates Battery State of Charge [%]
42         function SOC = GetBatterySOC(obj)
43             SOC = (obj.Battery_Level/obj.Battery_Capacity)*100;
44         end
45
46         function out = GetAverageSOC(obj)

```

```

44     out = 100*obj.Battery_Average/obj.Battery-Capacity;
45 end
46
47 % Update Battery Level and Array
48 function UpdateBattery(obj,time,Input,Disturb,offset)
49     % Account for Battery Saturation and Update Level
50     obj.Battery_Level = obj.UpdateBatteryCharge(Input,obj.Battery_Level,...
obj.Battery-Capacity);
51
52     if Disturb
53         obj.Battery_Level = obj.Battery_Level + offset;
54     end
55
56     % Update Battery Level Array
57     obj.Battery_Array(time) = obj.Battery_Level;
58
59     % Update the daily SOC average
60     obj.UpdateAverageSOC(time)
61 end
62
63 function UpdateAverageSOC(obj,time)
64
65     if time < obj.Average_Window+1
66         obj.Battery_Average = obj.Battery_Level;
67     else
68         len_batt = length(obj.Battery_Array());
69
70         if len_batt ≤ obj.Average_Window
71             start = 1;
72         else
73             start = len_batt-obj.Average_Window;
74         end
75
76         temp = obj.Battery_Array(start:len_batt);
77         obj.Battery_Average = mean(temp);
78
79     end
80     obj.Battery_Average_Array(time) = obj.Battery_Average;
81 end
82
83 % If the battery falls below the minimum threshold, return a false

```

```

84     % statement that will indicate that sensor data not be collected
85     % until the battery is charged up past the maximum threshold
86     function bool = WatchDog(obj)
87         persistent result;
88         % If result is empty matrix, populate it to default true.
89         if isempty(result)
90             result = true;
91         % When the max threshold is reached, start returning true
92         elseif obj.GetBatterySOC() ≥ obj.Battery.Watchdog_MaxLevel
93             result = true;
94         % If the battery falls below threshold, return false until it
95         % rises above maximum threshold
96         elseif obj.GetBatterySOC() ≤ obj.Battery.Watchdog_MinLevel
97             result = false;
98         end
99
100        bool = result;
101    end
102
103    %***Private Function*** Sets Battery Watchdog limits. Called in constructor
104    function Init_WatchDog(obj, SOC_Min, SOC_Max)
105        obj.Battery.Watchdog_MinLevel = SOC_Min;
106        obj.Battery.Watchdog_MaxLevel = SOC_Max;
107    end
108
109
110    %***Private Function*** Accounts for battery saturation
111    function Battery_Level_out = UpdateBatteryCharge(obj, Net_in, Battery_Level_in, ...
Battery_Capacity)
112        % Inputs: InputCharge: Amount of energy available to charge Battery
113        % Output: NONE: Internally adjusts Battery Level Property
114
115        %Define a Saturation Point
116        SaturationLevel = 0.9*Battery_Capacity;
117
118        if Net_in ≤ 0 %Adjust Battery Level if deficit
119            Battery_Level_out = Battery_Level_in + Net_in;
120        elseif Net_in > 0 % Charge the Battery
121
122            if Battery_Level_in ≥ SaturationLevel % Charge Saturation exists for ...
all of the input current

```

```

123         Battery_Level_out = Battery_Level_in + Net_in*(1+(-1)*(...
Battery_Level_in-SaturationLevel)/(Battery-Capacity-SaturationLevel));
124         elseif obj.Battery_Level + Net_in ≥ SaturationLevel; % Charge ...
Saturation Exists for part of the input current
125             % Charge to Saturation Point
126             ChargeToSat = SaturationLevel - Battery_Level_in;
127             Battery_Level_in = Battery_Level_in + ChargeToSat; % Adjust ...
Input Level
128             Net_in = Net_in - ChargeToSat; % Adjust Charge Level
129
130             % Take remaining Charge and apply saturation scaling to get ...
output
131             Battery_Level_out = Battery_Level_in + Net_in*(1+(-1)*(...
Battery_Level_in-SaturationLevel)/(Battery-Capacity-SaturationLevel));
132
133         else % No Charge Saturation Exists, Simply Add the Charge to the ...
battery
134             Battery_Level_out = Battery_Level_in + Net_in;
135         end
136     end
137
138     if Battery_Level_out ≤ 0 % Check for Battery out of lower bound
139         Battery_Level_out = 0;
140     end
141
142     % Check to make sure MaxLevel isn't exceeded
143     if Battery_Level_out ≥ Battery-Capacity
144         Battery_Level_out = Battery-Capacity;
145         return;
146     end
147 end
148
149     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
150     %                               End Methods
151     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
152 end % End Methods
153 end % End Classdef

```

---

## Appendix C

# Controller Script

Appendix C provides source code for a variable data collection rate system that used a discrete, time domain PID equation to control battery SOC by manipulating a GPS data collection rate. Gain scheduling is included for two sets of gains that can be specified by the user. Below is the source code for GetFixRate.m

---

```

1 function [New.FixRate, state] = GetFixRate(FirstCall, Old.FixRate,...
    Controller.Call.Period, SOC.F, Set.Point,Min.Fix.Rate,threshold,GS,Kp-,Kd-,...
    Ki-,Kp2,Kd2,Ki2)
2
3 persistent integral_err;
4 persistent prev_err;
5
6 if isempty(integral_err) && isempty(prev_err) || FirstCall
7     integral_err = 0;
8     prev_err = 0;
9 end
10
11 err = (Set.Point - SOC.F);
12
13 % Gain Scheduling
14
15 % When the error from the sp becomes sufficiently large, the set of
16 % gains for fast tracking to the set point should be used.
17
18 % If the error is sufficiently small (within a threshold) switch to a

```

```
19     % set of conservative gain values in order to still control but
20     % have less of an impact on the Fix rate.
21
22     if GS
23         if abs(err) > threshold
24             state = 1;
25             Kp = Kp_;%0.4;
26             Ki = Ki_;%0;
27             Kd = Kd_;%5000;
28             integral_err = 0;
29         else
30             state = 2;
31             Kp = Kp2; %0.1;
32             Ki = Ki2;%0;
33             Kd = Kd2;%2000;
34             integral_err = integral_err + err;
35         end
36     else
37         state = 1;
38         Kp = Kp_;%0.4;
39         Ki = Ki_;%0;
40         Kd = Kd_;%5000;
41         integral_err = integral_err + err;
42     end
43
44     dFR = round(Kp*err + Ki*integral_err*Controller_Call.Period + Kd*(err - ...
45     prev_err)/Controller_Call.Period);
46
47     prev_err = err;
48
49     New_FixRate = Old_FixRate + dFR;
50
51     if New_FixRate < Min_Fix_Rate
52         New_FixRate = Min_Fix_Rate;
53     elseif New_FixRate > 3600
54         New_FixRate = 3600;
55     end
56 end
```

---

# Bibliography

- [1] Jason Hsu, Sadaf Zahedi, Aman Kansal, Mani Srivastava, and Vijay Raghunathan. Adaptive duty cycling for energy harvesting systems. In *Proceedings of the 2006 international symposium on Low power electronics and design*, pages 180–185. ACM, 2006.
- [2] Fuller MR. Murray DL. A critical review of the effects of marking on the biology of vertebrates. *Research Techniques in Animal Ecology: Controversies and Consequences*, page 15–64, 2000.
- [3] Nils Warnock, John Y Takekawa, and Mary Anne Bishop. Migration and stopover strategies of individual Dunlin along the Pacific coast of North America. *Canadian Journal of Zoology*, 82(11):1687–1697, November 2004. ISSN 0008-4301, 1480-3283. doi: 10.1139/z04-154. URL <http://www.nrcresearchpress.com/doi/abs/10.1139/z04-154>.
- [4] Andrew J. Laughlin, Caz M. Taylor, David W. Bradley, Dayna LeClair, Robert G. Clark, Russell D. Dawson, Peter O. Dunn, Andrew Horn, Marty Leonard, Daniel R. Sheldon, Dave Shutler, Linda A. Whittingham, David W. Winkler, and D. Ryan Norris. Integrating information from geolocators, weather radar, and citizen science to uncover a key stopover area of an aerial insectivore. *The Auk*, 130(2):230–239, April 2013. ISSN 00048038, 19384254. doi: 10.1525/auk.2013.12229. URL <http://www.bioone.org/doi/abs/10.1525/auk.2013.12229>.
- [5] Juan M. Morales, Paul R. Moorcroft, Jason Matthiopoulos, Jacqueline L. Frair, John G. Kie, Roger A. Powell, Evelyn H. Merrill, and Daniel T. Haydon. Building the bridge between animal movement and population dynamics. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 365(1550):2289–2301, 2010. ISSN 0962-8436. doi: 10.1098/rstb.2010.0082.
- [6] Willem Bouten, Edwin W. Baaij, Judy Shamoun-Baranes, and Kees C. J. Camphuysen. A flexible GPS tracking system for studying bird behaviour at multiple scales. *Journal*

- of Ornithology*, 154(2):571–580, April 2013. ISSN 2193-7192, 2193-7206. doi: 10.1007/s10336-012-0908-1. URL <http://link.springer.com/10.1007/s10336-012-0908-1>.
- [7] Vishwas Raj Jain, Ravi Bagree, Aman Kumar, and Prabhat Ranjan. : GPS based animal tracking system. pages 617–622. IEEE, December 2008. ISBN 978-1-4244-2956-1, 978-1-4244-3822-8. doi: 10.1109/ISSNIP.2008.4762058. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4762058>.
- [8] Eli S. Bridge, Kasper Thorup, Melissa S. Bowlin, Phillip B. Chilson, Robert H. Diehl, René W. Fléron, Phillip Hartl, Roland Kays, Jeffrey F. Kelly, W. Douglas Robinson, and Martin Wikelski. Technology on the Move: Recent and Forthcoming Innovations for Tracking Migratory Birds. *BioScience*, 61(9):689–698, September 2011. ISSN 00063568, 15253244. doi: 10.1525/bio.2011.61.9.7. URL <http://bioscience.oxfordjournals.org/cgi/doi/10.1525/bio.2011.61.9.7>.
- [9] Andrew C Markham and Andrew J Wilkinson. Ecolocate: a heterogeneous wireless network system for wildlife tracking. In *Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics*, pages 293–298. Springer, 2008.
- [10] Clemens Moser, Jian-Jia Chen, and Lothar Thiele. Power management in energy harvesting embedded systems with discrete service levels. page 413. ACM Press, 2009. ISBN 9781605586847. doi: 10.1145/1594233.1594338. URL <http://portal.acm.org/citation.cfm?doid=1594233.1594338>.
- [11] Jason Hsu, Sadaf Zahedi, Aman Kansal, Mani Srivastava, and Vijay Raghunathan. Adaptive duty cycling for energy harvesting systems. page 180. ACM Press, 2006. ISBN 1595934626. doi: 10.1145/1165573.1165616. URL <http://portal.acm.org/citation.cfm?doid=1165573.1165616>.
- [12] Michael C. Hansen and Robert A. Riggs. Accuracy, Precision, and Observation Rates of Global Positioning System Telemetry Collars. *Journal of Wildlife Management*, 72(2):518–526, February 2008. ISSN 0022-541X, 1937-2817. doi: 10.2193/2006-493. URL <http://www.bioone.org/doi/abs/10.2193/2006-493>.
- [13] Hussein EL Ghor, Maryline Chetto, and Rafic Hage Chehade. A real-time scheduling framework for embedded systems with environmental energy harvesting. *Computers & Electrical Engineering*, 37(4):498–510, July 2011. ISSN 00457906. doi: 10.1016/j.compeleceng.2011.05.003. URL <http://linkinghub.elsevier.com/retrieve/pii/S0045790611000723>.



- [14] Dexin Li and Pai H Chou. Application/architecture power co-optimization for embedded systems powered by renewable sources. In *Proceedings of the 42nd annual Design Automation Conference*, pages 618–623. ACM, 2005.
- [15] Aman Kansal, Jason Hsu, Sadaf Zahedi, and Mani B. Srivastava. Power management in energy harvesting sensor networks. *ACM Transactions on Embedded Computing Systems*, 6(4):32–es, September 2007. ISSN 15399087. doi: 10.1145/1274858.1274870. URL <http://portal.acm.org/citation.cfm?doid=1274858.1274870>.
- [16] Christopher M. Vigorito, Deepak Ganesan, and Andrew G. Barto. Adaptive Control of Duty Cycling in Energy-Harvesting Wireless Sensor Networks. pages 21–30. IEEE, June 2007. ISBN 1-4244-1268-4. doi: 10.1109/SAHCN.2007.4292814. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4292814>.
- [17] Clemens Moser, Lothar Thiele, Davide Brunelli, and Luca Benini. Robust and Low Complexity Rate Control for Solar Powered Sensors. pages 230–235. IEEE, March 2008. ISBN 978-3-9810801-3-1, 978-3-9810801-4-8. doi: 10.1109/DATE.2008.4484691. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4484691>.
- [18] S. Peng and C.P. Low. Prediction free energy neutral power management for energy harvesting wireless sensor nodes. *Ad Hoc Networks*, 13:351–367, February 2014. ISSN 15708705. doi: 10.1016/j.adhoc.2013.08.015. URL <http://linkinghub.elsevier.com/retrieve/pii/S1570870513001789>.
- [19] David Anthony, William P. Bennett, Mehmet C. Vuran, Matthew B. Dwyer, Sebastian Elbaum, Anne Lacy, Mike Engels, and Walter Wehtje. Sensing through the continent: towards monitoring migratory birds using cellular sensor networks. page 329. ACM Press, 2012. ISBN 9781450312271. doi: 10.1145/2185677.2185747. URL <http://dl.acm.org/citation.cfm?doid=2185677.2185747>.
- [20] Raja Jurdak, Philipp Sommer, Branislav Kusy, Navinda Kottege, Christopher Crossman, Adam Mckeown, and David Westcott. Camazotz: multimodal activity-based GPS sampling. page 67. ACM Press, 2013. ISBN 9781450319591. doi: 10.1145/2461381.2461393. URL <http://dl.acm.org/citation.cfm?doid=2461381.2461393>.
- [21] Danielle D. Brown, Scott LaPoint, Roland Kays, Wolfgang Heidrich, Franz Kümmeth, and Martin Wikelski. Accelerometer-informed GPS telemetry: Reducing the trade-off between

- resolution and longevity. *Wildlife Society Bulletin*, 36(1):139–146, March 2012. ISSN 19385463. doi: 10.1002/wsb.111. URL <http://doi.wiley.com/10.1002/wsb.111>.
- [22] Pei Zhang, Christopher M. Sadler, Stephen A. Lyon, and Margaret Martonosi. Hardware design experiences in ZebraNet. page 227. ACM Press, 2004. ISBN 1581138792. doi: 10.1145/1031495.1031522. URL <http://portal.acm.org/citation.cfm?doid=1031495.1031522>.
- [23] Adam E. Duerr, Tricia A. Miller, Michael Lanzone, David Brandes, Jeff Cooper, Kieran O’Malley, Charles Maisonneuve, Junior A. Tremblay, and Todd Katzner. Flight response of slope-soaring birds to seasonal variation in thermal generation. *Functional Ecology*, pages n/a–n/a, December 2014. ISSN 02698463. doi: 10.1111/1365-2435.12381. URL <http://doi.wiley.com/10.1111/1365-2435.12381>.
- [24] Andrew Markham. *On a wildlife tracking and telemetry system: a wireless network approach*. PhD thesis, Department of Electrical Engineering, University of Cape Town, 2008.
- [25] ABE Jun-ichi, TAKADA Jun-ichi, Masatoshi DENDA, and Akira AKEYAMA. Real-time location estimation system for wild animals. *Institute of Electrical Engineers of Measurement Study Group document*, 2006(16):37–40, 2006.
- [26] William W. Cochran and Jr. Lord, Rexford D. A radio-tracking system for wild animals. *The Journal of Wildlife Management*, 27(1):pp. 9–24, 1963. ISSN 0022541X. URL <http://www.jstor.org/stable/3797775>.
- [27] R. Kays, S. Tilak, M. Crofoot, T. Fountain, D. Obando, A. Ortega, F. Kuemmeth, J. Mandel, G. Swenson, T. Lambert, B. Hirsch, and M. Wikelski. Tracking Animal Location and Activity with an Automated Radio Telemetry System in a Tropical Rainforest. *The Computer Journal*, 54(12):1931–1948, November 2011. ISSN 0010-4620, 1460-2067. doi: 10.1093/comjnl/bxr072. URL <http://comjnl.oxfordjournals.org/cgi/doi/10.1093/comjnl/bxr072>.
- [28] WILLIAM W Cochran, MARTIN Wikelski, R Greenberg, and PP Marra. Individual migratory tactics of new world catharus thrushes. *Birds of two worlds: the ecology and evolution of migration*. Johns Hopkins University Press, Baltimore, Maryland, USA, pages 274–289, 2005.

- [29] Ronald P Larkin, Arlo Raim, and Robert H Diehl. Performance of a non-rotating direction-finder for automatic radio tracking (desempeño de un localizador no-rotativo para radiotelemetría automática). *Journal of Field Ornithology*, pages 59–71, 1996.
- [30] L David Mech and Shannon M Barber. A critique of wildlife radio-tracking and its use in national parks. *Biological Resources Management Division, US National Park Service, Fort Collins, CO Technical Report*, 2002.
- [31] Charles R Tucker, Thomas A Radzio, Jeramie T Strickland, Ed Britton, David K Delaney, and Day B Ligon. Use of automated radio telemetry to detect nesting activity in ornate box turtles, terrapene ornata. *The American Midland Naturalist*, 171(1):78–89, 2014.
- [32] Adam S Hadley and Matthew G Betts. Tropical deforestation alters hummingbird movement patterns. *Biology Letters*, pages rsbl-2008, 2009.
- [33] Melissa S Bowlin, Per Henningsson, Florian T Muijres, Roel HE Vleugels, Felix Liechti, and Anders Hedenstrom. The effects of geolocator drag and weight on the flight ranges of small migrants. *Methods in Ecology and Evolution*, 1(4):398–402, 2010.
- [34] Hill RD. Theory of geolocation by light levels. *Le Boeuf BJ, Laws RM, eds. Elephant Seals: Population ecology, behaviour, and physiology*, page 227–236, 1994.
- [35] Erich Bächler, Steffen Hahn, Michael Schaub, Raphaël Arlettaz, Lukas Jenni, James W. Fox, Vsevolod Afanasyev, and Felix Liechti. Year-Round Tracking of Small Trans-Saharan Migrants Using Light-Level Geolocators. *PLoS ONE*, 5(3):e9566, March 2010. ISSN 1932-6203. doi: 10.1371/journal.pone.0009566. URL <http://dx.plos.org/10.1371/journal.pone.0009566>.
- [36] Adam M. Fudickar, Martin Wikelski, and Jesko Partecke. Tracking migratory songbirds: accuracy of light-level loggers (geolocators) in forest habitats: Accuracy of geolocators. *Methods in Ecology and Evolution*, 3(1):47–52, February 2012. ISSN 2041210X. doi: 10.1111/j.2041-210X.2011.00136.x. URL <http://doi.wiley.com/10.1111/j.2041-210X.2011.00136.x>.
- [37] Eli S. Bridge, Jeffrey F. Kelly, Andrea Contina, Richard M. Gabrielson, Robert B. MacCurdy, and David W. Winkler. Advances in tracking small migratory birds: a technical review of light-level geolocation: Light-Level Geolocation Dataloggers. *Journal of Field Ornithology*, 84(2):121–137, June 2013. ISSN 02738570. doi: 10.1111/jfo.12011. URL <http://doi.wiley.com/10.1111/jfo.12011>.

- [38] M. Hebblewhite and D. T. Haydon. Distinguishing technology from biology: a critical review of the use of GPS telemetry data in ecology. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1550):2303–2312, July 2010. ISSN 0962-8436, 1471-2970. doi: 10.1098/rstb.2010.0087. URL <http://rstb.royalsocietypublishing.org/cgi/doi/10.1098/rstb.2010.0087>.
- [39] Elliott D. Kaplan and C. Hegarty, editors. *Understanding GPS: principles and applications*. Artech House mobile communications series. Artech House, Boston, 2nd ed edition, 2006. ISBN 1580538940.
- [40] S. M. Tomkiewicz, M. R. Fuller, J. G. Kie, and K. K. Bates. Global positioning system and associated technologies in animal behaviour and ecological research. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1550):2163–2176, July 2010. ISSN 0962-8436, 1471-2970. doi: 10.1098/rstb.2010.0090. URL <http://rstb.royalsocietypublishing.org/cgi/doi/10.1098/rstb.2010.0090>.
- [41] R. Bajaj, S.L. Ranaweera, and D.P. Agrawal. Gps: location-tracking technology. *Computer*, 35(4):92–94, Apr 2002. ISSN 0018-9162. doi: 10.1109/MC.2002.993780.
- [42] Eli S. Bridge, Kasper Thorup, Melissa S. Bowlin, Phillip B. Chilson, Robert H. Diehl, René W. Fléron, Phillip Hartl, Roland Kays, Jeffrey F. Kelly, W. Douglas Robinson, and Martin Wikelski. Technology on the Move: Recent and Forthcoming Innovations for Tracking Migratory Birds. *BioScience*, 61(9):689–698, September 2011. ISSN 00063568, 15253244. doi: 10.1525/bio.2011.61.9.7. URL <http://bioscience.oxfordjournals.org/cgi/doi/10.1525/bio.2011.61.9.7>.
- [43] B Naef-Daenzer, D Fruh, M Stalder, P Wetli, and E Weise. Miniaturization (0.2 g) and evaluation of attachment techniques of telemetry transmitters. *Journal of Experimental Biology*, 208(21):4063–4068, 2005.
- [44] Philip John Seddon and Richard Maloney. *Tracking wildlife radio-tag signals by light fixed-wing aircraft*. Department of Conservation Wellington, New Zealand, 2004.
- [45] Russell W Bradley, Fred Cooke, Lynn W Loughheed, and W Sean Boyd. Inferring breeding success through radiotelemetry in the marbled murrelet. *Journal of Wildlife Management*, 68(2):318–331, 2004.
- [46] B. J. M. Stutchbury, S. A. Tarof, T. Done, E. Gow, P. M. Kramer, J. Tautin, J. W. Fox, and V. Afanasyev. Tracking Long-Distance Songbird Migration by Using Geolocators. *Science*,

- 323(5916):896–896, February 2009. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1166664. URL <http://www.sciencemag.org/cgi/doi/10.1126/science.1166664>.
- [47] Erin M. Kennedy, Joshua R. Kemp, Corey C. Mosen, George L. W. Perry, and Todd E. Dennis. GPS telemetry for parrots: A case study with the Kea ( *Nestor notabilis* ). *The Auk*, 132(2):389–396, April 2015. ISSN 0004-8038, 1938-4254. doi: 10.1642/AUK-14-196.1. URL <http://www.bioone.org/doi/abs/10.1642/AUK-14-196.1>.
- [48] J. M. Igual, M. G. Forero, G. Tavecchia, J. Gonzalez-Solis, A. Martinez-Abraín, K. A. Hobson, X. Ruiz, and D. Oro. Short-term effects of data-loggers on corys shearwater (*Calonectris diomedea*). *Marine Biology*, 146(3):619–624, February 2005. ISSN 0025-3162, 1432-1793. doi: 10.1007/s00227-004-1461-0. URL <http://link.springer.com/10.1007/s00227-004-1461-0>.
- [49] Gabriel E. Machovsky-Capuska, Mark E. Hauber, Mariela Dassis, Eric Libby, Martin C. Wikelski, Rob Schuckard, David S. Melville, Willie Cook, Michelle Houston, and David Raubenheimer. Foraging behaviour and habitat use of chick-rearing Australasian Gannets in New Zealand. *Journal of Ornithology*, 155(2):379–387, April 2014. ISSN 2193-7192, 2193-7206. doi: 10.1007/s10336-013-1018-4. URL <http://link.springer.com/10.1007/s10336-013-1018-4>.
- [50] de Jongh A Mira A Boitani L Quaglietta L, Martins BH. A low-cost gps gsm/gprs telemetry system: Performance in stationary field tests and preliminary data on wild otters (*lutra lutra*). *PLoS ONE*, 2012. doi: 10.1371/journal.pone.0029235.
- [51] Roland Kays, Patrick A. Jansen, Elise M.H. Knecht, Reinhard Vohwinkel, and Martin Wikelski. The effect of feeding time on dispersal of *Virola* seeds by toucans determined from GPS tracking and accelerometers. *Acta Oecologica*, 37(6):625–631, November 2011. ISSN 1146609X. doi: 10.1016/j.actao.2011.06.007. URL <http://linkinghub.elsevier.com/retrieve/pii/S1146609X1100107X>.
- [52] Moe Rahnema. Overview of the gsm system and protocol architecture. *Communications Magazine, IEEE*, 31(4):92–100, 1993.
- [53] David J. Anthony, William P. Bennett, Mehmet C. Vuran, Matthew B. Dwyer, Sebastian Elbaum, and Felipe Chavez-Ramirez. Simulating and testing mobile wireless sensor networks. page 49. ACM Press, 2010. ISBN 9781450302746. doi: 10.1145/1868521.1868531. URL <http://portal.acm.org/citation.cfm?doid=1868521.1868531>.

- [54] Martin Wikelski, Roland W Kays, N Jeremy Kasdin, Kasper Thorup, James A Smith, and George W Swenson. Going wild: what a global small-animal tracking system could do for experimental biologists. *Journal of Experimental Biology*, 210(2):181–186, 2007.
- [55] Danielle D Brown, Roland Kays, Martin Wikelski, Rory P Wilson, and A Peter Klimley. *Observing the unwatchable through acceleration logging of animal behavior*. Bibliothek der Universität Konstanz, 2014.
- [56] Christopher M Vigorito, Deepak Ganesan, and Andrew G Barto. Adaptive control of duty cycling in energy-harvesting wireless sensor networks. In *Sensor, Mesh and Ad Hoc Communications and Networks, 2007. SECON'07. 4th Annual IEEE Communications Society Conference on*, pages 21–30. IEEE, 2007.
- [57] Trong Nhan Le, Olivier Sentieys, Olivier Berder, Alain Pegatoquet, and Cecile Belleudy. Power Manager with PID Controller in Energy Harvesting Wireless Sensor Networks. pages 668–670. IEEE, November 2012. ISBN 978-1-4673-5146-1, 978-0-7695-4865-4. doi: 10.1109/GreenCom.2012.107. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6468387>.
- [58] Kong Soon Ng, Chin-Sien Moo, Yi-Ping Chen, and Yao-Ching Hsieh. Enhanced coulomb counting method for estimating state-of-charge and state-of-health of lithium-ion batteries. *Applied energy*, 86(9):1506–1511, 2009.
- [59] Y Çadırcı and Y Özkazanç. Microcontroller-based on-line state-of-charge estimator for sealed lead–acid batteries. *Journal of power sources*, 129(2):330–342, 2004.
- [60] K.J. Åström and T. Hägglund. The future of PID control. *Control Engineering Practice*, 9(11):1163–1175, November 2001. ISSN 09670661. doi: 10.1016/S0967-0661(01)00062-4. URL <http://linkinghub.elsevier.com/retrieve/pii/S0967066101000624>.
- [61] Ed. Piscataway Levine, William S. "PID Control" in *The control handbook*.
- [62] Mikael Pohjola. Pid controller design in networked control systems. *Helsinki University of Technology*, 2006.
- [63] Kiam Heong Ang, G. Chong, and Yun Li. PID control system analysis, design, and technology. *IEEE Transactions on Control Systems Technology*, 13(4):559–576, July 2005. ISSN 1063-6536. doi: 10.1109/TCST.2005.847331. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1453566>.

- [64] Gui-Ying Yang and Julia Slingo. The diurnal cycle in the tropics. *Monthly Weather Review*, 129(4):784–801, 2001.
- [65] Spilios Theodoulis and Gilles Duc. Gain-scheduled autopilot synthesis for an atmosphere re-entry vehicle. In *AIAA Guidance, Navigation, and Control Conference and Exhibit. Honolulu, USA*, 2008.
- [66] Wilson J Rugh and Jeff S Shamma. Research on gain scheduling. *Automatica*, 36(10):1401–1425, 2000.
- [67] Vibha Prasad and Sang H Son. Classification of analysis techniques for wireless sensor networks. pages 93–97, 2007.
- [68] Geoff V. Merrett, Neil M. White, Nick R. Harris, and Bashir M. Al-Hashimi. Energy-Aware Simulation for Wireless Sensor Networks. pages 1–8. IEEE, June 2009. ISBN 978-1-4244-2907-3. doi: 10.1109/SAHCN.2009.5168932. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5168932>.
- [69] Shuo Pang, Jay Farrell, Jie Du, and Matthew Barth. Battery state-of-charge estimation. In *American Control Conference, 2001. Proceedings of the 2001*, volume 2, pages 1644–1649. IEEE, 2001.
- [70] Nicolas Watrin, Benjamin Blunier, and Abdellatif Miraoui. Review of adaptive systems for lithium batteries state-of-charge and state-of-health estimation. In *Transportation Electrification Conference and Expo (ITEC), 2012 IEEE*, pages 1–6. IEEE, 2012.
- [71] Jun Xu, Binggang Cao, Zheng Chen, and Zhongyue Zou. An online state of charge estimation method with reduced prior battery testing information. *International Journal of Electrical Power & Energy Systems*, 63:178–184, 2014.
- [72] Mehdi Gholizadeh and Farzad R Salmasi. Estimation of state of charge, unknown nonlinearities, and state of health of a lithium-ion battery based on a comprehensive unobservable model. *Industrial Electronics, IEEE Transactions on*, 61(3):1335–1344, 2014.
- [73] Wen-Yeau Chang. The state of charge estimating methods for battery: a review. *International Scholarly Research Notices*, 2013, 2013.
- [74] Davide Andrea. State of charge estimate with li-ion batteries. November 2009. URL [http://liionbms.com/php/wp\\_soc\\_estimate.php](http://liionbms.com/php/wp_soc_estimate.php).

- [75] Christina V. Schwarz, Brian J. Reiser, Elizabeth A. Davis, Lisa Kenyon, Andres Achér, David Fortus, Yael Shwartz, Barbara Hug, and Joe Krajcik. Developing a learning progression for scientific modeling: Making scientific modeling accessible and meaningful for learners. *Journal of Research in Science Teaching*, 46(6):632–654, 2009. ISSN 1098-2736. doi: 10.1002/tea.20311. URL <http://dx.doi.org/10.1002/tea.20311>.
- [76] Bruno Cargnelutti, Aurélie Coulon, A. J. Mark Hewison, Michel Goulard, Jean-Marc Angibault, and Nicolas Morellet. Testing global positioning system performance for wildlife monitoring using mobile collars and known reference points. *The Journal of Wildlife Management*, 71(4):pp. 1380–1387, 2007. ISSN 0022541X. URL <http://www.jstor.org/stable/4496205>.
- [77] I Buchman. Batteries in a portable world: A handbook on rechargeable batteries for non-engineers. *Cadex Electronics Inc., Richmond*, 2001.
- [78] Electronics-lab. How to rebuild a li-ion battery pack. [http://www.electronics-lab.com/articles/Li\\_Ion\\_reconstruct/index.html](http://www.electronics-lab.com/articles/Li_Ion_reconstruct/index.html).