Idiographic Models of Walking Behavior for Personalized mHealth Interventions: Some Novel Approaches

by

Rachael T. Kha

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved July 2022 by the
Graduate Supervisory Committee:

Daniel E. Rivera, Chair
Shuguang Deng
Christopher Muhich

ARIZONA STATE UNIVERSITY

August 2022

ABSTRACT

This thesis presents the development of idiographic models (i.e., single subject or N = 1) of walking behavior as a means of facilitating the design of control systems to optimize mobile health (mHealth) interventions for sedentary adults. Model-on-Demand (MoD), an adaptive modeling technique, is demonstrated as an ideal method for modeling nonlinear systems with noise on a simulated continuously stirred tank reactor (CSTR). Comparing MoD to AutoRegressive with eXogenous input (ARX) estimation, MoD outperforms ARX in terms of addressing both nonlinearity and noise in the CSTR system. With the CSTR system as an initial proof of concept, MoD is then used to model individual walking behavior using intervention data from participants of HeartSteps, a walking intervention that studies the effect of within-day suggestions. Given the number of possible measured features from which to design the MoD models, as well as the number of model parameters that influence the model's performance, optimizing MoD models through exhaustive search is infeasible. Consequently, a discrete implementation of simultaneous perturbation stochastic approximation (DSPSA) is shown to be an efficient algorithm to find optimal models of walking behavior. Combining MoD with DSPSA, models of walking behavior were developed using participant data from Just Walk, a day-to-day walking intervention; MoD outperformed ARX models on both estimation and validation data. DSPSA was also applied to ARX modeling, highlighting the use of DSPSA to not only search over model parameters and features but also data partitioning, as DSPSA was used to evaluate models under various combinations of estimation and validation data from a single participant's walking data. Results of this thesis point to ARX with DSPSA as a routine means for dynamic model estimation in large-scale behavioral intervention settings.

ACKNOWLEDGEMENTS

From my first time being a UGTA in Fall 2018 to completing my master's degree in Summer 2022, and many classes and applications in-between, Dr. Daniel E. Rivera has been one of the most constant sources of support and encouragement throughout my entire academic career. While there is a lot that I am thankful for, taking CHE 461 with Dr. Rivera was one of the most important turning points in my life. It was the first time I felt really inspired by any of my engineering classes and probably the main reason why I am still in engineering today. Although I will not be continuing in the Control Systems Engineering Lab for my PhD, I will probably be thanking Dr. Rivera in every acknowledgments section that I write from here on out.

I am also grateful to the many students and professors that I had the pleasure of working with during my master's research, including the other CSEL students, Mohamed, Sarasij, and Owais, as well as everyone from the U01 and NLM meetings, especially Dr. Eric Hekler and Dr. Pedja Klasnja who not only helped me write my first, first-authored paper, but also gave me lots of advice about grad school. Though nearly all of our interactions were virtual, I always felt like I was part of a supportive network of researchers. One of the best parts of the weekly meetings was getting to see so many people with such diverse interests and experiences collaborate on novel research, which helped me feel confident in bringing my own diverse background into research. And I also enjoyed the many philosophical tangents, animal interruptions, and rants about current events. They were as entertaining as they were informative. I also appreciate the effort and feedback of Dr. Shuguang Deng and Dr. Christopher Muhich, who served on my thesis committee.

I am also thankful to my parents and grandparents, who helped pave the way for me to complete my bachelor's and master's degrees. As a third generation engineering student and second generation ASU graduate, I am grateful for their support and patience, especially as I spent a lot of time jumping between disciplines and had to take many summer classes to stay on track. I know that none of us really had any idea what I was trying to accomplish (myself included), but I'm glad it seemed to work out better than we expected.

Support for this research has been provided by the National Institutes of Health (NIH) through grants R01LM013107 and U01CA229445.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

NOMENCLATURE

The following list defines several acronyms used throughout this thesis.

AIC Akaike Information Criterion

AICN Akaike Information Criterion according to Loader's Definition

ARX AutoRegressive with eXogenous Input

CSTR Continuously Stirred Tank Reactor

DSPSA Discrete Simultaneous Perturbation Stochastic Approximation

GCV Generalized Cross Validation

GoF Goodness of Fit criterion

mHealth Mobile Health

MoD Model-on-Demand

MPC Model Predictive Control

NRMSE Normalized Root Mean Squared Error

PA Physical Activity

SCT Social Cognitive Theory

SPSA Simultaneous Perturbation Stochastic Approximation

Chapter 1

INTRODUCTION

The unprecedented availability of data made possible today by advances in technology and increased use of mobile devices has allowed dynamic modeling to become a primary source of data-driven solutions to problems in many fields, including behavioral medicine. However, moving from data to dynamical models presents many challenges, as the informative utility of data is limited by its ability to be operationalized in both an explanatory and predictive sense. This means that special attention must be given to (1) experimental design for obtaining quality data and (2) estimating useful models that inform decision-making in interventions. This thesis focuses on the latter.

Despite the availability of data (or the relative ease by which we are able to collect data), distilling a comprehensive and useful model often requires exhaustive search and high computational power. These challenges arise from (1) the large number of measured features that are potential model inputs and (2) the presence of noise. To more efficiently find optimal models, this thesis demonstrates the use of discrete simultaneous perturbation stochastic approximation (DSPSA) to optimize models of individual walking behavior. DSPSA is a simulation-based technique that optimizes models through stochastic search (Spall, 1998; Wang and Spall, 2014). It is especially useful when a closed-form objective function is not available and when measurements may be noisy. Here, the results of DSPSA as a tool for feature selection, model order selection, and parameter estimation, are illustrated, which will allow users to more easily obtain models from large volumes of data and utilize more computationally-demanding models. This will be shown thorough the use and optimization of Model-on-Demand (MoD) and AutoRegressive with eXogenous (ARX) input existimation. The former is an appealing approach for modeling noisy, nonlinear systems (Stenman, 1999; Braun *et al.*, 2001) and the latter is a classical and commonly used dynamic modeling technique.

Behavioral medicine presents a rich field of opportunities to study novel modeling methods as there has been a push for idiographic (i.e., "single subject") approaches to understand each individual's specific barriers to health-promoting behavior and personalized behavior-change interventions. While the benefits of increased physical activity (PA) are well-documented, there remains an issue in how to promote *sustained* increases. Providing individuals with information about improving their health through increased PA, alone, is not enough to benefit at-risk populations. Significant improvement requires that participants both respond to suggestions to increase PA and sustain engagement at higher levels of PA over extended periods of time, beyond that of an intervention. Personalized interventions designed from idiographic models may be able to bridge these gaps by directly delivering helpful suggestions to increase physical activity within the particular individual's day and better their attitude and habits towards physical activity to create sustained improvements over time. However, the framework for idiographic modeling is still being developed, as it has not yet been explored as deeply as nomothetic approaches (i.e., to find generalizable knowledge or scientific laws), which have been widely practiced across many fields.

HeartSteps and *Just Walk* are interventions that address physical inactivity through individualized physical activity (PA) interventions delivered to participants via mobile health (mHealth) technologies. Both aim to also study the dynamical nature of behavior to better account for psychosocial, contextual, and environmental factors and how the effects of these change with time, to improve upon traditional theories of behavior change (Spruijt-Metz *et al.*, 2022). *HeartSteps* is a microrandomized trial that studies within-day decisions on participants' walking behavior. Participants were sent up to five contextually-tailored activity suggestions per day, and their walking behavior following the suggestions was measured (Klasnja *et al.*, 2019). Combining control systems theory, behavioral science, and informatics, *Just Walk* utilized system identification principles to design goal setting and positive reinforcement models (Phatak *et al.*, 2018; Freigoun *et al.*, 2017; Rivera *et al.*, 2018). Through this transdisciplinary approach, *Just Walk* not only highlights the improvements from taking an individualized approach, but also demonstrates the benefit of system identification principles in the development of personalized interventions. These studies both account for the complexity of physical activity and its dependence on environmental and mental factors, addressing the importance of idiographic modeling in overcoming the limitations of nomothetic approaches, which are the current dominant paradigm, to successfully model individual participant behavior.

An individual's walking behavior is complex and idiosyncratic, influenced by a variety of factors that themselves may be context dependent. To develop a useful model of behavior, it is necessary to capture this complexity and nonlinearity. While dynamic models for behavior have been developed with linear AutoRegressive with eXogenous input (ARX) modeling, these may be too simple to provide sufficient explanation and prediction of an individual's PA behavior, which is necessary to produce an effective intervention. The simplicity of ARX restricts its capacity to capture nonlinearity in individual behavior and may not allow for prediction or explanation of behavior as an individual's environment or mental state changes. Consequently, MoD presents an appealing approach, as it fits a local model at each operating point, allowing it more flexibility and adaptability to build models "on demand." While MoD may provide a better method to model PA, it is more computationally-demanding than ARX and requires greater prior knowledge, as there are additional parameters that need to be specified. However, this increase in the complexity of implementation can be mitigated by DSPSA, making MoD more accessible and applicable in a broader set of application settings.

One drawback of using a small data approach to behavioral medicine, is that there is limited data that can be used for modeling, as there are constraints to the amount of data that researchers are able to obtain from individual participants. A specific issue that arises from this, is that the choice of estimation and validation data can influence the accuracy of the model, especially in its ability to predict an individuals future behavior under different conditions. For example, using the first half of an individual's data to estimate a model may result in a model that does not capture certain influences such as notification fatigue. This model might then overestimate the individual's response to future intervention notifications, resulting in suggestions with unattainable goals that discourage the individual from participating in the intervention. To address this, DSPSA will also be used to optimize and validate models given varying sets of estimation and validation data formed by combining different segments of an individual's walking behavior. Since partitioning an individual's data into estimation and validation data presents another parameter that might influence the success of a model, it is another opportunity to demonstrate DSPSA as an optimization method.

The results presented in this thesis contribute to a more recent development of control-oriented approaches to the study of behavioral systems and the design of behavioral interventions. Prior research have demonstrated that control systems engineering provides a useful framework to develop and optimize individualized interventions that adapt to a person's evolving needs and environment. Using mathematical models of physical behavior, a controller can be designed to provide decision rules that govern when and what kinds of walking suggestions (i.e. the magnitude of a step goal) are provided to intervention participants (Hekler *et al.*, 2018). Aside from the controller itself, control systems principles can be used in all parts of intervention development, including data gathering. One example of this is the use of a control-oriented dynamical systems model of social cognitive theory (SCT) based on fluid analogies, which was used for system identification (Martin *et al.*, 2018). However, the research demonstrated in this thesis contributes specifically to developing idiographic models from an individual's data with respect to (1) modeling technique and (2) the scalability of model specification. As mentioned previously, behavioral systems are nonlinear and noisy, as human behavior is highly contextual and difficult to measure (i.e. variability in self-reported measures, step count errors, etc.). To address this, we demonstrate that Model-on-Demand estimation can better handle both nonlinearity and noise than ARX estimation. With a modeling technique in mind, however, there still remains the issue of feature selection and parameter specification, which can be tedious depending on the number of measured variables available for model inputs. This problem grows as we consider that models will need to be specified many individuals, since the interventions are personalized. DSPSA is proposed as an optimization technique to quickly and efficiently specify parameters and select features that are most important to the individual's walking behavior. This makes the process of defining an individual's model routine and efficient so that it can be performed on a much larger scale.

This thesis is organized as follows: Chapter 2 presents the advantages of MoD over ARX, demonstrated on both a simulated continuously-stirred tank reactor (a nonlinear system with noise) and *HeartSteps*, a walking intervention that studies within-day decisions. Chapter 3 combines MoD with discrete simultaneous perturbation stochastic approximation (DSPSA) to optimize behavioral models with respect to feature selection and parameter specification, using individual walking data provided by the *Just Walk* study. Chapter 4 addresses the issue of partitioning individual participants' walking data into estimation and validation data by applying DSPSA to ARX models obtained and tested with different sets of estimation and validation data. Chapter 5 summarizes the results and contributions of this thesis, as well as potential future work.

1.1 Publication Summary

Research from this thesis that has been published thus far as refereed conference papers are listed below.

[1] Kha, R. T., Rivera, D. E., Klasnja, P., Hekler, E. (2022), "Model Personalization in Behavioral Interventions using Model-on-Demand Estimation and Discrete Simultaneous Perturbation Stochastic Approximation," in *Proceedings of 2022 American Control Conference (ACC)*, Altanta, pp. 671-676.

Chapter 4 is being prepared for submissions to the 2023 American Control Conference in San Diego.

Chapter 2

MODEL-ON-DEMAND ESTIMATION

2.1 Model-on-Demand Overview

Model-on-Demand (MoD) is a hybrid modeling technique that combines both local and global modeling. At each operating point, MoD fits a local model by solving a weighted regression problem. Unlike global models, which fit a model to the entire estimation data set and then discard the data, Model-on-Demand does not discard the data set after a model is defined. Instead, MoD uses a subset of the data to fit a local model and returns to a stored estimation data set, repeating the process, as it computes local models at each operating point. The neighborhood size of the data used is adjusted at each operating point to minimize the local model's error, making MoD an adaptive modeling technique (Stenman, 1999).

One of the primary advantages of Model-on-Demand, compared to traditional global estimation methods, is that the model is optimized locally. MoD applies a weighted regression to generate local estimates, adjusting the neighborhood size from a stored database of observations to build models 'on demand.' This optimizes the bias/variance trade-off locally, allowing MoD to achieve lower errors for a fixed model structure. This is particularly useful as the number of observations increases, as it becomes more difficult to obtain a global model that is optimal over an entire data set (Stenman, 1999). MoD is also advantageous, as it is simple to use in selecting a local model and computes estimates relatively fast.

The MoD modeling formulation can be described with a single input single output (SISO) process as demonstrated by the approach of Braun *et al.* (2001). Consider a

SISO process with nonlinear ARX structure,

$$y(t) = m(\varphi(t)) + e(t) \tag{2.1}$$

in which $m(\cdot)$ is an unknown nonlinear mapping and $e(t)$ is an error term. The error is modeled as a random signal with zero mean and variance σ_k^2. The MoD predictor attempts to estimate output predictions, $\hat{y}(i)$, based on a local neighborhood of the regressor space $\varphi(t)$. The regressor vector takes on the form of a linear ARX structure as shown in Equation 2.2.

$$\varphi(t) = [y(t-1)\dots y(t-n_a)\ u(t-n_k)\dots u(t-n_b-n_k)]^T \tag{2.2}$$

where n_a denotes the number of previous lags in the output, n_b denotes the number of previous lags in the input, and n_k denotes the delay in the model. Note that for multi-output systems, n_a is specified for each output, and for a multi-input system, n_b and n_k are specified for each input. The applications used to demonstrate MoD in later parts of this thesis are multi-input single output.

A local estimate $\hat{y}(i)$ is then obtained at each operating point from the solution of the weighted regression problem, shown in Equation 2.3:

$$\hat{\beta} = \arg\min_{\hat{\beta}} \sum_{i=1}^{N} \ell(y(i) - \hat{m}(\varphi(i),\ \hat{\beta})) \times W\left(\frac{\|\varphi(i)-\varphi(t)\|_M}{h}\right) \tag{2.3}$$

in which $\ell(\cdot)$ is a quadratic norm function, $\|u\|_M \triangleq \sqrt{u^T M u}$ is a scaled distance function on the regressor space, h is a bandwidth parameter controlling the size of the local neighborhood, and $W(\cdot)$ is a window function (also referred to as the kernel) assigning weights to each remote data point based on its distance from $\varphi(t)$ (Braun *et al.*, 2001). N is the total number of observations available for model estimation, and the weights control the "locality" of the data points, influencing the size of the local neighborhood for each operating point. The window is typically a bell-shaped function with bounded support. MoD users specify k_{min}, k_{max}, and a goodness-of-fit criterion, which then affect the bandwidth parameter and the window function. Two common goodness-of-fit criteria include the Akaike Information Criterion (AIC) and Generalized Cross Validation (GCV), but many others may also be considered (Stenman, 1999). Assuming a local model structure,

$$m(\varphi(t), \beta) = \beta_0 + \beta_1^T(\varphi(i) - \varphi(t)) \tag{2.4}$$

which is linear in the unknown parameters, a MoD estimate can be computed using least squares methods. Denoting β_0 and β_1 as the minimizers of Equation 2.3 using the model from Equation 2.4, a one-step ahead prediction is given by

$$\hat{y}(i) = \alpha + \beta_1^T\varphi(i) \tag{2.5}$$

where $\alpha = \beta_0 - \beta_1^T\varphi(t)$. Each local regression problem produces a single prediction $\hat{y}(i)$ corresponding to the current regression vector $\varphi(t)$. To obtain a prediction at other operating points in the regressor space, MoD adapts both the relative weights and the selection of data to optimize a new local model at the next operating point. This diverges from global modeling techniques in which the model is estimated from the data once, and then the data is discarded. The bandwidth h, which is computed adaptively at each prediction, controls the neighborhood size governing the trade-off between the bias and variance errors of the estimated model.

In application, users can specify the regressor structure used in $\varphi(t)$, the local polynomial order that approximates $m(\cdot)$, k_{min}, k_{max}, and the goodness-of-fit criterion; these variables impact the size of the neighborhood chosen to fit the local model. The window function is also user-specified, but in most cases (including the results presented in this thesis), a tricube window function is used, since its derivative is continuous and it goes to zero at the boundaries (Braun, 2001).

The MoD model is then evaluated using Normalized Root Mean Square Error (NRMSE). The formulation is defined in Equation 2.6, in which θ represents the model parameters, $\hat{y}$ is the model output, y is the actual value of the data, and $\bar{y}$ is the average of the data. A good model has higher fit percentages ($J(\theta)$), in which a perfect model has a fit of 100%.

$$J(\theta) = 100\% \times \left(1 - \frac{||y - \hat{y}||_2}{||y - \bar{y}||_2}\right) \tag{2.6}$$

This fit is distinct from the goodness-of-fit criterion used in the MoD formulation (i.e. GCV, AIC, AICN, etc.), as the goodness-of-fit used in the MoD algorithm is used to evaluate the minimization problem, which calculates the local model at each operating point. The NRMSE fit is used to evaluate the MoD model as a whole. The NRMSE fit will be used throughout this thesis to evaluate both MoD and ARX models.

2.2 Continuously-Stirred Tank Reactor Application

To demonstrate the advantages of Model-on-Demand (MoD) in a well-defined setting, MoD was compared to a linear AutoRegressive with eXogenous input (ARX) model in its ability to handle both nonlinearity and noise with respect to a simulated continuously-stirred tank reactor (CSTR) as per Bequette (1998). The CSTR system model evaluated here was created in Simulink (see Appendix A). Using an exothermic reaction with reactant A producing product B,

$$A \longrightarrow B \tag{2.7}$$

the CSTR simulation assesses the influence of jacket temperature as the manipulated variable available for control. This is shown in Figure 2.1. The feed stream is a source of disturbances, in terms of both temperature (T_f), and the inlet concentration of A (C_{Af}). Two outputs of interest were measured: (1) tank temperature T and (2) outlet concentration C_A.

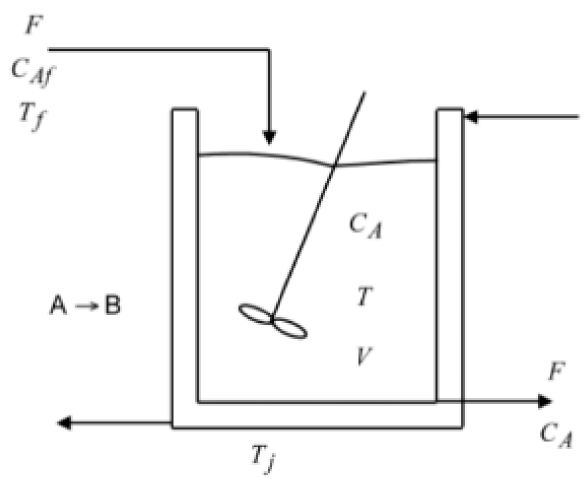

Figure 2.1: Continuously Stirred Tank Reactor (reprinted)

The CSTR simulation, was also modified to account for both process and measurement noise. Noise was introduced by adding values produced by a random number generator with zero mean and nonzero variance. Various noise levels were implemented and used to compare MoD to a simpler modeling technique, namely ARX. The noise level was controlled by adjusting the variance of the number generator (i.e. larger variance, higher level of noise).

To run the simulation, the output variables of interest were subjected to changes in the jacket temperature, as per a multi-sinusoidal (multisine) signal designed using the Input Design GUI (Bailey and Rivera, 2020). The multisine signal was designed such that there was persistent excitation (see Appendix B). From the open loop response (Figure 2.2), the dominant time constant was specified with a low value of 3 hours and a high value of 5 hours, as determined from the response of the outlet stream (C_A).

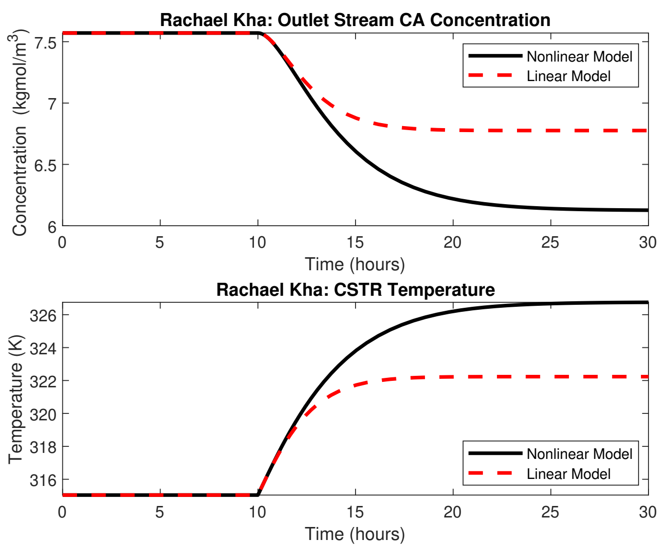

Figure 2.2: CSTR Open Loop Response to a Step Increase (5°C) in Jacket Temperature

Separate estimation and validation datasets were obtained using distinct realizations of the same input signal (Figure 2.3). These signals were designed with an amplitude of ±1. When implemented in the simulation, this signal was multiplied by a user-specified gain and added to the initial jacket temperature to provide sufficient variation to produce data for modeling. Each realization was run through the simulation, and the responses of the output variables were measured. The input and output dynamics were then used to test both MoD and ARX estimation. Responses from one signal were reserved for estimation data, while responses from the second realization were used as validation data. Additional information is available in Appendix B.

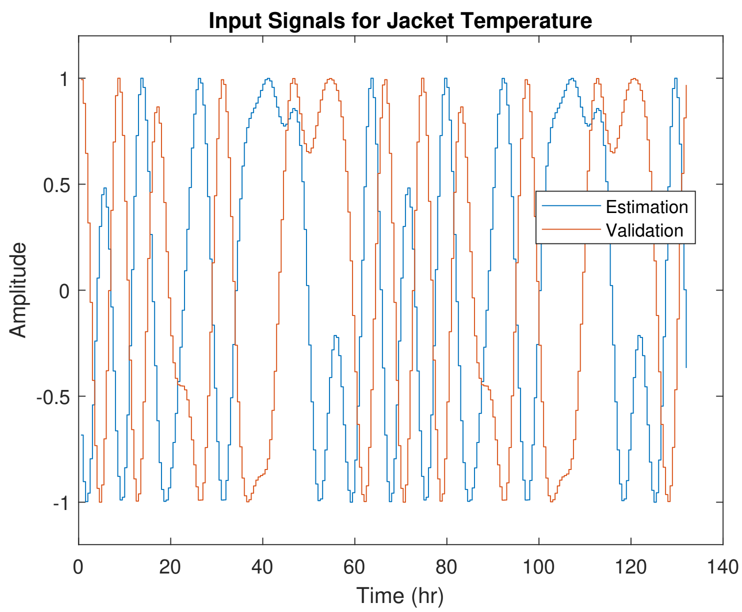

Figure 2.3: CSTR: Input Signals for Jacket Temperature to Produce Estimation and Validation Data

The results of the CSTR case study are summarized in Table 2.1. They demonstrate MoD's ability to handle both nonlinearity and noise better than ARX. Despite increases in noise, MoD is able to obtain a higher fit to the validation data and the MoD model also achieves this with lower RMS and maximum error. Although two controlled variables are measured (C_A, T), only output concentration was be used to evaluated MoD. More detail about these results are also presented in the next sections of this chapter.

	NRMSE Fit (%)		RMS Error ($kgmol/m^3$)		Max Error ($kgmol/m^3$)	
	MoD	ARX	MoD	ARX	MoD	ARX
Noise-Free	94.56	71.27	0.0723	0.3821	0.2190	0.7083
Low Noise	65.39	47.85	0.5758	0.8677	2.3028	3.3736
High Noise	13.10	8.52	1.3974	1.4712	3.8587	3.9620

Table 2.1: CSTR Simulation: MoD and ARX Results Summary

In Table 2.1, the noise cases refer to differences in measurement noise introduced to the outputs of interest. In Simulink, this was modeled by adding a random number generator to the calculated values. The level of noise was then controlled by specifying the variance of the random number generator. For “low noise,” the variance was set to 0.001, while for “high noise,” the variance was set to 0.1. The “noise free” case does not have any measurement noise.

2.2.1 CSTR, Noise-Free Conditions

In the noise-free case, MoD's ability to better model nonlinearity than ARX is highlighted in Figure 2.4. Neither process nor measurement noise were implemented. This is also a single input single output (SISO) system, in which the measured output is concentration of A and the input is jacket temperature. Both disturbances were considered at steady-state.

The modeling results from this simulation were obtained using the MoD parameter specifications outlined in Table 2.2. The ARX regressor structure was used common to both the MoD and ARX models.

Parameter	Value
Polynomial order	1
ARX Regressor Structure, $[n_a\ n_b\ n_k]$	[2 2 1]
Goodness of Fit	AICN
k_{min}, k_{max}	55, 320

Table 2.2: CSTR: Noise Free Model Parameters

From Figure 2.4, MoD achieved a higher percent fit to the validation data, measured in terms of normalized root mean squared error (NRMSE), at $\approx 94\%$. This is about 23% higher than the NRMSE fit obtained by ARX estimation, which is $\approx 71\%$. The difference in performance between MoD and ARX is also illustrated by the root mean square error (RMS) and the maximum error (MAX), which are both lower for MoD.

From the middle and bottom subplots, additional features of the MoD estimation model are shown. The middle subplot illustrates the neighborhood size used to estimate the local model at each operating point, which varies between k_{min} =55, and

k_{max} =320. Most local models were computed in the vicinity of 300 data points for estimation, but at a few operating points, MoD used less, about 200 data points.

The difference between the MoD predicted output and the actual system output (in terms of output concentration) at each operating point is demonstrated in the bottom subplot. There are four large peaks, at which MoD underestimates the output concentration ($y - y_{est} > 0$), but the difference between the MoD predicted and actual outputs is no more than 0.25 $kgmol/m^3$.

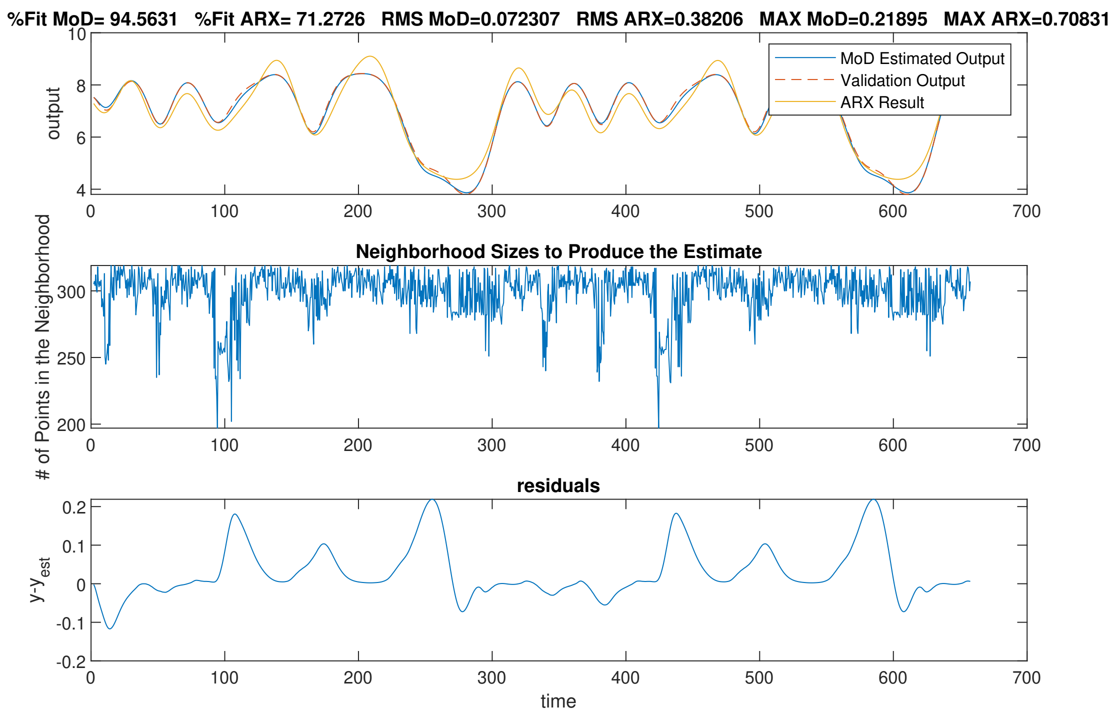

Figure 2.4: CSTR: Noise-Free MoD and ARX Estimation. Output is C_A, measured in $kgmol/m^3$.

2.2.2 CSTR, Low Noise Conditions

The low noise case was assessed by running the CSTR simulation with noisy input and output signals as per the variance values in Table 2.3.

Measurement Noise Signal	Variance
Tank Temperature	0.001
Outlet Concentration	0.001
Process Noise Signal	**Variance**
Jacket Temperature	0
Feed Concentration	10
Feed Temperature	10

Table 2.3: CSTR: Low Noise Process and Measurement Noise Variance

Using the MoD parameters outlined in Table 2.4 and the ARX orders, [2 2 1], the results in Figure 2.5 were obtained.

Parameter	Value
Polynomial order	1
ARX Regressor Structure, $[n_a \ n_b \ n_k]$	[2 2 1]
Goodness of Fit	AIC
k_{min}, k_{max}	55, 500

Table 2.4: CSTR: Low Noise Model Parameters

From Figure 2.5, once again MoD provided a better model than ARX, measured in terms of NRMSE percentage fit. The MoD model had a fit of $\approx$ 65%, while ARX resulted in a fit of $\approx$ 48%, which is a difference of about 17%. MoD's ability to

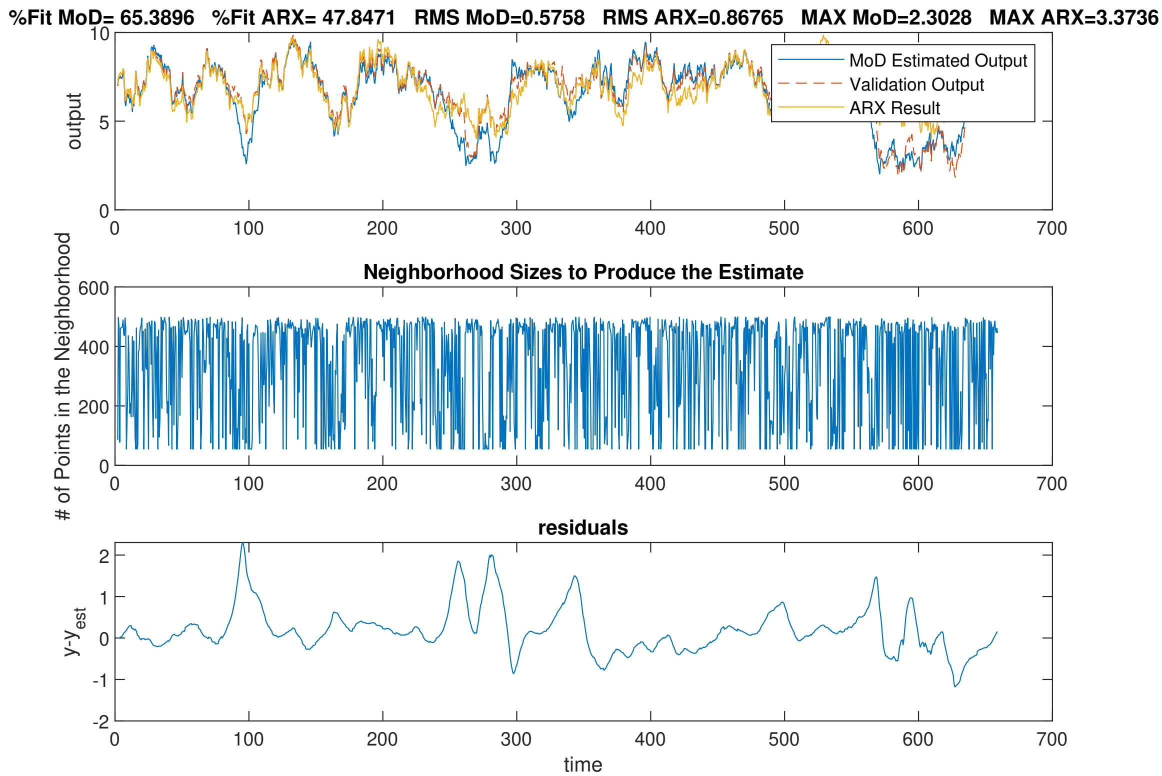

Figure 2.5: CSTR: Low Noise MoD and ARX Estimation. Output is C_A, measured in $kgmol/m^3$.

better predict the concentration of A in the output stream of the CSTR system is also highlighted by its lower RMS error, 0.5758, and lower maximum error, 2.3028. ARX had a RMS error of 0.8677 and a maximum error of 3.3736.

Unlike the noise-free case, under low noise, the size of the neighborhood used to estimate a MoD model at each operating point displays a lot more variability, using between 150 and 500 data points. The magnitude of the residuals were also larger than in the noise-free case.

2.2.3 CSTR, High Noise Conditions

In the high noise case, the measurement noise for the tank and output concentrations were increased by increasing the variance of the random number generator by two orders of magnitude from the low noise case (0.1 versus 0.001). Process noise was kept the same between the two cases.

Measurement Noise Signal	Variance
Tank Temperature	0.1
Outlet Concentration	0.1
Process Noise Signal	Variance
Jacket Temperature	0
Feed Concentration	10
Feed Temperature	10

Table 2.5: CSTR: High noise Process and Measurement Noise Variance

Parameter	Value
Polynomial order	1
ARX Regressor Structure, $[n_a\ n_b\ n_k]$	[2 2 1]
Goodness of Fit	GCV
k_{min}, k_{max}	55, 320

Table 2.6: CSTR: High Noise Model Parameters

Both the fit provided by MoD and ARX are lower in the high noise case than the low noise case, which was expected (Figure 2.6). However, MoD still provides a better fit than ARX, althought the difference here is about 5%, unlike prior cases where the difference was closer to 20%. In terms of the RMS error and maximum

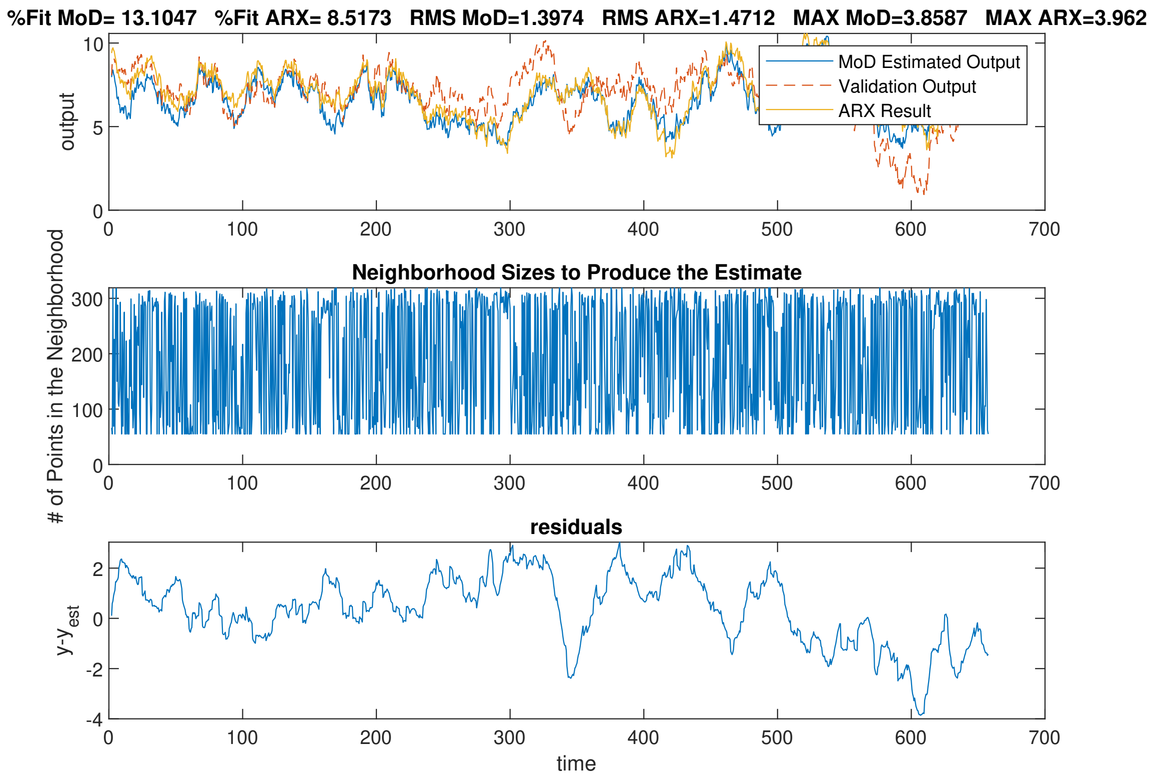

Figure 2.6: CSTR: High Noise MoD and ARX Estimation. Output is C_A, measured in $kgmol/m^3$.

error, the difference between the two models is also smaller than in previous cases. However, based on these measures, MoD still performs better than ARX, since the errors calculated from the MoD model are smaller than those from the ARX model. This degradation in MoD's outperforming ARX as noise increases is likely due to noise dominating the system, rather than nonlinearity. One of MoD's primary advantages is its ability to model nonlinearity, as it is an adaptive method that fits a local model at each operating point. However, as noise increases in the system, the modeling complications that arise from nonlinearity become dominated by complications due to noise, and so using MoD over ARX has diminishing benefits at higher noise levels.

Compared to the low noise case, the high noise case also has larger residuals and there is a lot of variance in the size of the neighborhood used to estimate a model at each operating point.

From the three CSTR cases presented, MoD outperforms ARX despite both no-linearity and noise, which suggests that it may be more appropriate than ARX to model other nonlinear, noisy systems such as human behavior. As such, MoD and ARX will also be assessed in terms of their ability to model walking behavior for developing personalized interventions.

2.3 Behavioral Intervention Applications

In this thesis, data from two behavioral intervention studies will be used to demonstrate the use of Model-on-Demand (MoD) as a method to estimate idiographic models of walking behavior. Both studies, *HeartSteps* and *Just Walk*, aim to provide sedentary adults with personalized interventions to increase physical activity. To do this, both study the impact of walking suggestions sent to participants via mobile health devices, such as a Fitbit and mobile apps. These suggestions are tailored to individuals, accounting for local environmental factors such as weather and time of day, as well as idiosyncratic factors such as the individual's mental state, which may influence the individual's responsiveness and therefore the effectiveness of the study. These interventions are designed to be dynamic and adapt to changes in real-time, to maximize the impact of each suggestion. Both studies collected data of the participants' environmental context and mental state to study the participants' responsiveness to the intervention, measured in terms of the participants' step count. These data are then used in this chapter to illustrate the effectiveness of Model-on-Demand as a modeling technique to understand individuals' walking behavior and inform the interventions that provide walking suggestions. In later chapters of this thesis, the same behavioral data will also be used to assess Discrete Simultaneous Perturbation Stochastic Approximation (DSPSA) to optimize models developed from participants' individual data. In the following sections *HeartSteps* data is used to compare MoD with ARX estimation and show that MoD is better suited for behavioral interventions. *Just Walk* will be discussed in the next chapter.

2.3.1 *HeartSteps* Overview

The first study, *HeartSteps* studies the impact of within-day walking decisions. It is a micro-randomized trial used to inform just-in-time adaptive interventions (JITAI).

Variable	Description
z_{30}	log of step count in the last 30 minutes
z_{60}	log of step count in the last 60 minutes
d(t)	day of the decision point
d_{25}	number of notifications in the past 25 decision points
d_{30}	number of notifications in the past 30 decision points
A(t)	a suggestion (any) at decision point t (1 if any suggestion is sent)
A_1	binary (1 if active suggestion)
A_2	anti-sedentary suggestion (1 if anti-sedentary suggestion)
Home	binary (1 if location = home)
Work	binary (1 if location = work)
Other	binary (1 if location = other)
Weekday	binary (0 if weekday, 1 if weekend)
Weather	binary (1 if outdoor weather, 0 if indoor weather)
Morning	binary (1 if time = morning)
Afternoon	binary (1 if time = afternoon)
Mid-Day	binary (1 if time = mid-day)
Evening	binary (1 if time = evening)
Night	binary (1 if time = night)

Table 2.7: *HeartSteps*: Input Variable Definitions

The adaptive intervention framework accounts for the participant's current context (environmental and mental) to decide whether an intervention should be delivered and if so, what kind of intervention suggestion should be sent (Klasnja *et al.*, 2019). 'Micro-randomization' refers to trial structure of the study, which randomly assigns a treatment to participants. These treatments are the types of interventions, which vary between active suggestions and anti-sedentary suggestions. The former directly encourages walking, while the latter suggest general movement such as stretching or standing up and are sent to increase engagement with the intervention, reinforcing future responsiveness.

HeartSteps was a six week microrandomized trial in which participants received up to five interventions per day (from five intervention decision points), and 44 adults participated in the study. As such, there are about 200 data points per individual. The variables measured in this study are listed in Table 2.7. Only one output variable is measured: $y(t+1)$, which is the log of the individual's step count in the 30 minutes after decision point t. From the other measured variables, which are also potential features include in modeling, most relate to notifications (total amount and type), measured walking (step count in the 30 and 60 minutes prior to a notification), and environmental factors (location, time of day, and weather). Note that the *Weather* variable is not a measure of temperature, but instead 'indoor' or 'outdoor' weather.

2.3.2 *HeartSteps MoD Case Studies*

Model-on-Demand (MoD) was able to provide better idiographic models of walking behavior compared to ARX on data from the *HeartSteps* intervention. This is illustrated using walking data provided by Participant 7. This will further be demonstrated by three MoD implementations, which illustrate MoD's performance (relative to ARX) under various modeling parameters.

In these cast studies, 'estimation' results denote when both the estimation data and validation data are the same (the same dataset is used to estimate the model and to validate the model). This is different from 'prediction' results in which separate datasets were used for estimation and validation. Unlike the CSTR case, each participant only has one set of data, and so two distinct datasets are not available to be used as estimation and validation data. Consequently, for the predictive cases, a separate validation dataset was created by reserving a subset of the individual's walking data.

	NRMSE Fit (%)		RMS Error ($kgmol/m^3$)		Max Error ($kgmol/m^3$)	
	MoD	ARX	MoD	ARX	MoD	ARX
Case 1	54.74	24.78	1.359	2.258	5.793	6.182
Case 2	37.15	23.52	1.887	2.296	5.606	6.065
Case 3	35.64	32.98	1.780	1.855	4.965	4.878

Table 2.8: *HeartSteps*: MoD and ARX Comparisons for Participant 7

Three case studies will be shown using Participant 7 of the *HeartSteps* intervention (Figure 2.7). Each case study will compare MoD models with ARX models obtained under various modeling conditions. The first case compares the two techniques on the participant's whole data set, as used for both model estimation and validation.

The MoD and ARX models both use all measured variables as features. The second case compares the two modeling methods on the participant's whole data set (used as both estimation and validation data) given only a subset of features from the available measured variables. This demonstrates that MoD can still outperform ARX even with limited features. The last case split's the participant's data into estimation and validation, and compares MoD and ARX models on the validation set, while using a subset of variables as model features. The results from each case study are outlined in Table 2.8.

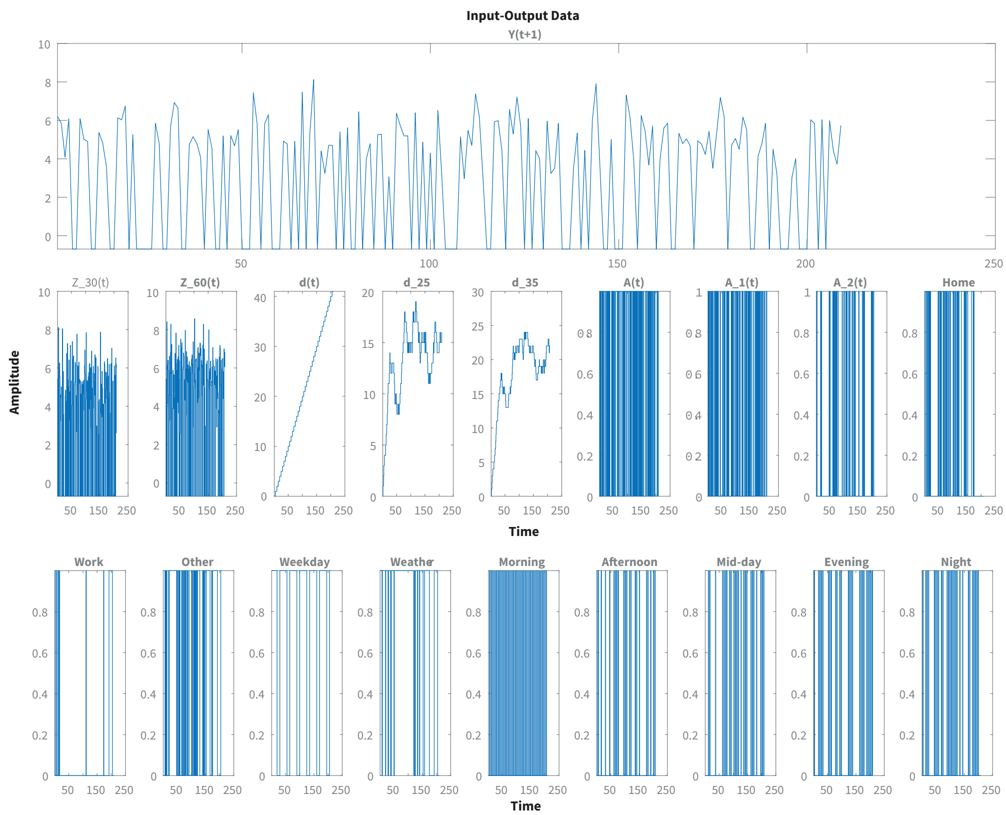

Figure 2.7: *HeartSteps*: Participant 7 Data

Case Study 1

In Figure 2.8, MoD and ARX were compared with respect to their ability to model a given participant's walking behavior. The results shown are with respect to estimation data – the data used to inform the model and the validate the model are the same. All 18 inputs were used, and the ARX orders for the models is [1 1 0] ([n_a n_b n_k]), in which the same n_b and n_k values were used for all inputs. The parameters are further outlined in Table 2.9 and the features are listed in Table 2.10.

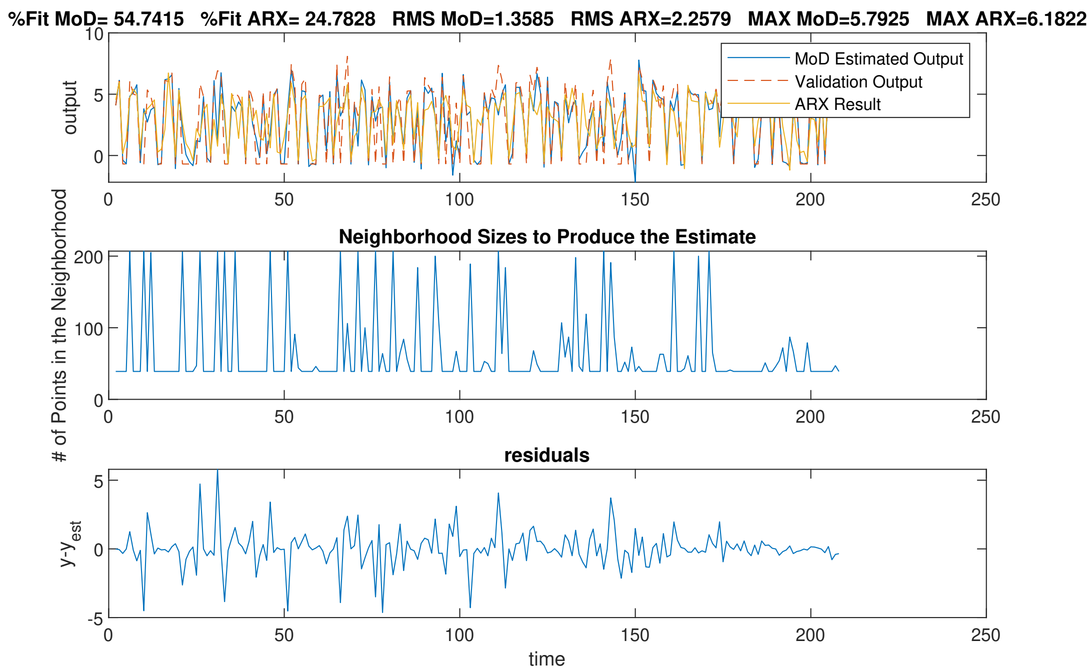

Figure 2.8: *HeartSteps*: Participant 7 MoD and ARX Model Comparison on Estimation Data with All Features

The MoD model produces a fit that is almost twice as high as that of the ARX model (54.7% versus 24.7%). The RMS and maximum MoD errors are also smaller than those of the ARX model. From the second subplot in Figure 2.8, the neighborhood size required to fit each local model is often small, around 40-50 data points.

Parameter	Value
Polynomial Order, P	1
ARX Regressor Structure, $[n_a\ n_b\ n_k]$	[1 1 0]
Goodness of Fit	AICN
k_{min}, k_{max}	5, 500

Table 2.9: *HeartSteps*: Model Parameters for Figure 2.8

Feature	Included in Model?	Feature	Included in Model?
z_{30}	X	Work	X
z_{60}	X	Other	X
d(t)	X	Weekday	X
d_{25}	X	Weather	X
d_{30}	X	Morning	X
A(t)	X	Afternoon	X
A_1	X	Mid-Day	X
A_2	X	Evening	X
Home	X	Night	X

Table 2.10: *HeartSteps*: Feature Selection for Figure 2.8

Case Study 2

Figure 2.9 demonstrates MoD's application to the same participant's walking behavior as in Figure 2.8, except the model uses a smaller subset of features. This case study further illustrates MoD's ability to model behavioral data, as the NRMSE fit from the MoD model still outperforms the ARX model on the estimation data by about 14%, even when using only 10 of the total 18 features. Both the RMS and maximum errors of the MoD model are also smaller than the errors from the ARX model. The parameters and features that lead to these results are specified in Tables 2.11 and 2.12 respectively.

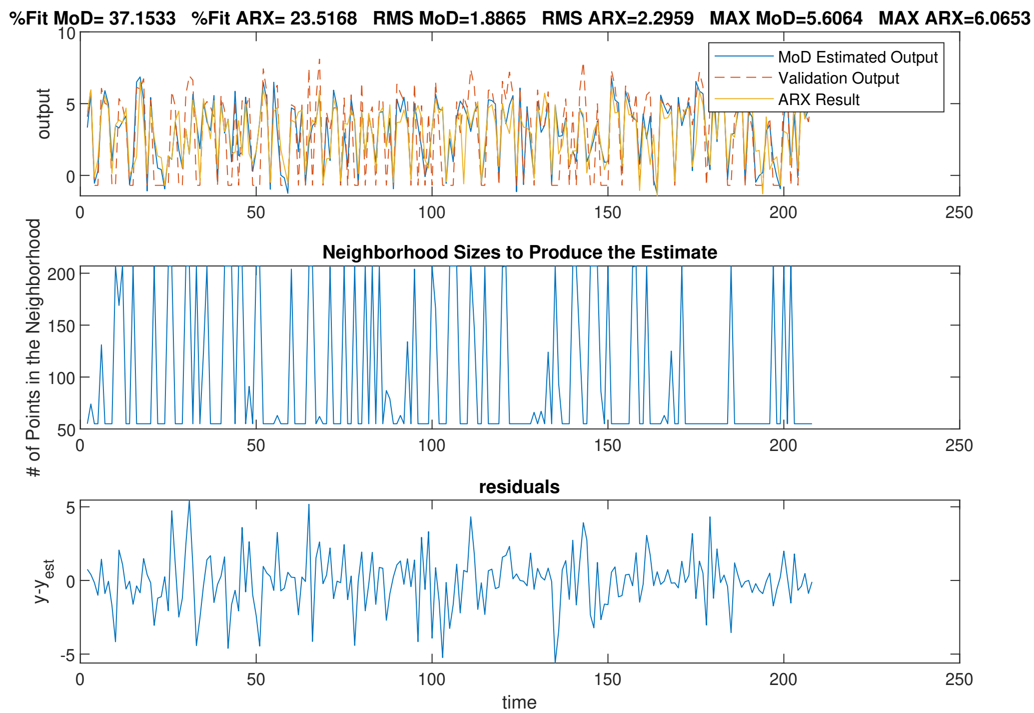

Figure 2.9: *HeartSteps*: Participant 7 MoD and ARX Model Comparison on Estimation Data with a Subset of Features (10), per Table 2.12

Although the model using all 18 features has a better fit than the model using only 10 features, including all features may not be necessary or possible, when developing models for personalized interventions. A smaller set of features may be required to reduce computational complexity. Also, including all features may lead to overfitting or be redundant.

Reducing the number of features further sheds light on the relative significance of each feature to the individual's walking behavior. Specifically, the features used

Parameter	Value
Polynomial Order, P	1
ARX Regressor Structure, $[n_a\ n_b\ n_k]$	[1 1 0]
Goodness of Fit	AICN
k_{min}, k_{max}	55, 500

Table 2.11: *HeartSteps*: Model Parameters for Figure 2.9

Feature	Included in Model?	Feature	Included in Model?
z_{30}	X	Work	-
z_{60}	-	Other	X
d(t)	-	Weekday	X
d_{25}	-	Weather	X
d_{30}	-	Morning	X
A(t)	-	Afternoon	X
A_1	X	Mid-Day	-
A_2	X	Evening	X
Home	X	Night	-

Table 2.12: *HeartSteps*: Feature Selection for Figure 2.9

in the reduced model play an important role in the individual's physical activity, and together provide some significant explanation of the individual's behavior. The individual contributions of each feature are unclear, and the contributions relative to the features *not* included in the model need further explanation. However, this demonstrates that all features may not be necessary. The threshold (i.e. in terms of NRMSE fit) by which a set of features may be evaluated may also need to be specified, but this will need to be determined by further studies.

Case Study 3

Using the same participant, Figure 2.10 illustrates the use of MoD and ARX in a *predictive* case, in which the models are evaluated on validation data rather than the same estimation data used to inform the models. In this case, the validation data was the last 25% of the participant's data, while the first 75% of their data was used as estimation data.

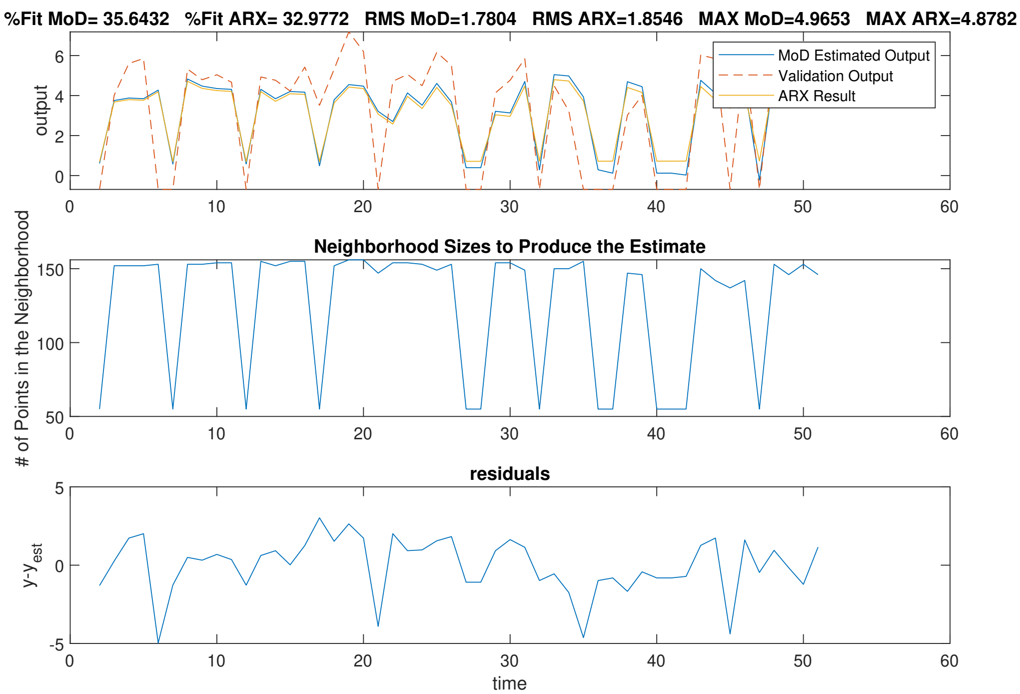

Figure 2.10: *HeartSteps*: Participant 7 MoD and ARX Model Comparison on Validation Data (25%) with a Subset of Features, per Table 2.13

After searching across many combinations of features, it was found that for the predictive case, the highest NRMSE fits (for both MoD and ARX) were obtained with smaller subsets of features (i.e. usually 3 features or less). This is the opposite relationship found with the estimation data, in which more features generally lead to a higher NRMSE fit. The results in Figure 2.10 were obtained using only *two features*: z_60 and $d(t)$, which denote the log of the step count in the prior 60 minutes and the day of the decision point, respectively. The MoD parameters are specified in Table 2.14.

Feature	Included in Model?	Feature	Included in Model?
z_{30}	-	Work	-
z_{60}	X	Other	-
d(t)	X	Weekday	-
d_{25}	-	Weather	-
d_{30}	-	Morning	-
A(t)	-	Afternoon	-
A_1	-	Mid-Day	-
A_2	-	Evening	-
Home	-	Night	-

Table 2.13: *HeartSteps*: Feature Selection for Figure 2.10

Parameter	Value
Polynomial Order, P	1
ARX Regressor Structure, $[n_a\ n_b\ n_k]$	[0 1 0]
Goodness of Fit	GCV
k_{min}, k_{max}	55, 500

Table 2.14: *HeartSteps*: Model Parameters for Figure 2.10

The difference between the MoD and ARX NRMSE fit is about 2%, which is a much narrower difference than in the previous cases on estimation data. For the predictive case, it was much more difficult to find parameters in which MoD provided a significantly better model. Of the set of parameters that were searched over, many resulted in MoD and ARX being comparable at best. However, only a limited set of possible combinations of features and parameters were assessed. An improved search mechanism is presented in the next chapter, in which simultaneous perturbation stochastic approximation (SPSA) is used to search over parameter values and features to find optimal MoD conditions.

Seeing as these results were found through exhaustive or "brute force" search over MoD parameters and feature selection, it is unclear whether there are sets of parameters and features that would lead to even better models, especially for the predictive case. But since there are so many combinations of both parameters and features to be searched over, exhaustive search seems infeasible or at least inefficient. Consequently, to better search over parameters, simultaneous perturbation stochastic approximation (SPSA) will be used to optimize the MoD models. Since many of the parameters use discrete values, and the feature selection can be viewed as a binary problem (1 if the feature is used in the model, 0 if it is not used), a discrete implementation of SPSA was used. This is shown in the next chapter.

2.4 Summary and Conclusions

Using Model-on-Demand to estimate models of both a simulated nonlinear CSTR with noise and a real participant's walking data from *HeartSteps*, obtained higher fit percentages than ARX, a traditionally-used global linear modeling technique. On the CSTR system, MoD models outperformed ARX models under both no noise and noisy conditions. This suggests that MoD is a better modeling method for nonlinear, noisy systems than ARX. MoD was then compared to ARX in behavioral systems, which are nonlinear and noisy, using data from *HeartSteps*. Since *HeartSteps* was an intervention performed with human participants, distinct estimation and validation data sets could not be obtained, so participants' data were split into the first 75% used as estimation data and the latter 25% used as validation data. MoD outperformed ARX on the estimation data and was marginally better than ARX on the validation data. However, the parameters used in these MoD models were defined by manually searching over both model parameters and features, so not all possible models were evaluated. As such, it is possible that other MoD models could have been obtained with different parameter values and features that significantly outperforms ARX models on both estimation and validation data. However, the model performance may also be limited due to characteristics of the data itself. Parameter and feature choice will be addressed in the next chapter, which discusses discrete simultaneous perturbation stochastic approximation (DSPSA) as a search mechanism. Nonetheless, the results demonstrated in this chapter show that MoD is an effective modeling technique for behavioral studies and produces models that perform comparably or better than ARX. The rest of this thesis will focus on methods to optimize idiographic models.

Chapter 3

MODEL-ON-DEMAND WITH SPSA

3.1 Motivations

While Model-on-Demand provides a method to estimate models for noisy, nonlinear dynamics, choosing the parameters that produce an optimal model presents a separate challenge. The MoD parameters include choices of local polynomial order, ARX structure, and goodness-of-fit criterion, in addition to the features or inputs used to estimate the model, giving rise to a large number of combinations, across which an exhaustive "brute force" search would be impractical. For example, a case with 18 features available, such as *HeartSteps*, would require a search over $2^{18} - 1 = 262,143$ combinations. Accounting for model parameters (i.e. regressor orders, etc.) increases the number of combinations even further. To estimate optimal model parameters and bypass the need for exhaustive search, we propose the use of a discrete form of Simultaneous Perturbation Stochastic Approximation (DSPSA), a simulation-based optimization technique. The results shown in later sections of this chapter were also published as a conference paper, presented the 2022 American Control Conference (Kha *et al.*, 2022).

SPSA is a popular technique that is useful in contexts where a closed-form objective function is not available and where noise may be present (Spall, 1998). It provides a non-deterministic approach to typical gradient descent methods. For feature selection, binary SPSA was found to outperform other methods, including Binary Genetic Algorithms, and conventional feature selection methods such as Sequential Forward Selection, Sequential Backward Selection, and Sequential Forward Floating Selection.

These methods were evaluated on multiple datasets, in which binary SPSA was found to perform at least comparably in small datasets (less than 100 features) and highly favorably in large datasets (over 100 features), as measured by cross-validation error (Aksakalli and Malekipirbazari, 2016).

3.2 SPSA Overview

To use SPSA, we start with a guess of the model parameter values $\hat{\theta}$, which are updated with each iteration. To obtain an estimated gradient, all model parameters are subjected to a random, two-sided simultaneous perturbation, which are then used to evaluate the objective function, $J(\hat{\theta})$. These two evaluations are then used to approximate the gradient, which is subsequently used to update the parameter values. This is repeated for a user-specified number of iterations k.

The objective function chosen for the optimization problem often takes on the form of a loss function, L, which is not readily available or explicit and can instead be approximated by noisy measurements, $J(\theta) = L(\theta) + \epsilon(\theta)$. SPSA then minimizes the loss function through a process that resembles gradient descent, iterating and updating θ.

SPSA has been used in many problems spanning a diverse set of fields, including supply chain management and public health (Wang and Spall, 2014; Schwartz *et al.*, 2006). As will be demonstrated in this thesis, SPSA is also useful in behavioral medicine, providing model parameter estimations and feature selection for individualized health interventions. SPSA can be used to search simultaneously across both continuous and discrete parameter values. Discrete SPSA (DSPSA) will be shown here, used to optimize both model parameters and feature selection.

The following summarizes the DSPSA process for k iterations, as described in Wang and Spall (2014) and Aksakalli and Malekipirbazari (2016):

1. *Initialize the Input Vector and Gain Sequences.* Specify an initial p-dimensional input vector, $\hat{\theta}$, in which p corresponds to the number of features or parameters subject to stochastic search. The gain sequences a_k and c_k define the step size of each iteration and perturbation, respectively.

2. *Generate the Perturbation Vector.* Generate a perturbation vector (Δ_k) of dimension p using a Bernoulli ± 1 distribution with probability $1/2$.

3. *Create Two Input Vectors for Gradient Approximation.* From the input vector, create a new vector, $\pi(\hat{\theta}_k) = \lfloor \hat{\theta}_k \rfloor + \mathbf{1}_p/2$, in which $\lfloor \cdot \rfloor$ is the floor operator and $\mathbf{1}_p$ is a p-dimensional vector of ones. From $\pi(\hat{\theta}_k)$, create two input vectors for gradient approximation, $\hat{\theta}_k^+ = \pi(\hat{\theta}_k) + c_k\Delta_k$ and $\hat{\theta^-}_k = \pi(\hat{\theta}_k) - c_k\Delta_k$. Apply bounds to limit between discrete values and round $\hat{\theta}_k^+$ and $\hat{\theta}_k^-$.

4. *Approximate the Gradient.* Evaluate the objective function $J(\cdot)$ at the bounded and rounded input vectors, θ_k^+ and θ_k^-. Use these two evaluations to approximate the gradient using a finite difference approximation:
$$\hat{g_k} = \frac{J(\hat{\theta}_k^+) - J(\hat{\theta}_k^-)}{2c_k\Delta_k} \tag{3.1}$$

5. *Update the Input Vector.* Using the gradient approximation, update the input vector:
$$\hat{\theta}_{k+1} = \hat{\theta}_k - a_k\hat{g_k} \tag{3.2}$$
Apply bounds to limit between discrete values and round the new input vector.

6. *Report the Best Solution Vector.* Once the SPSA search has reached its final iteration, report the best solution.

In a binary application of DSPSA (where the parameter choices are limited to only 0 and 1), it is necessary to correspondingly limit the input vector between 0 and 1 at each iteration (i.e. at steps 3 and 5).

3.3 Behavioral Applications: *Just Walk*

While the former chapter used data from *HeartSteps* to demonstrate the use of Model-on-Demand (MoD), this chapter will use *Just Walk* to illustrate how MoD models can not only be applied to behavioral interventions but also be optimized efficiently by using a stochastic search mechanism, discrete simultaneous perturbation stochastic approximation (DSPSA).

3.3.1 Just Walk Overview

The second study, *Just Walk* is a between-day intervention, in which individuals are sent a daily step count goal via mobile device, and the participants' actual step count, as well as other contextual variables (i.e. self-reported variables, environmental factors), are measured (Phatak *et al.*, 2018). *Just Walk* was designed using a system identification approach, instead of a microrandomized trial like in *HeartSteps*.

Variable	Definition
Step Goal	Pseudo-random Multisine Signal Centered around Baseline PA
Expected Points	Pseudo-random Multisine Signal, 100 to 500 points
Granted Points	Raw Score, 0 to 500
Weekday/Weekend	0 = weekday, 1 = weekend
Predicted Stress	1 - 5 scale (self-reported)
Predicted Busyness	1 - 4 scale (self-reported)
Predicted Typicality	1 - 4 scale (self-reported)

Table 3.1: *Just Walk*: Variable Definitions

Step goals and expected points (which are rewarded upon achieving their step goals), were designed as pseudo-random multisine signals. These were delivered to participants and used to study walking behavior as an open loop system. The point values translate to dollar amounts (100 points is \$0.20 and 500 points is \$1.00) awarded as \$5.00 electronic Amazon gift cards after every 2500 points. The total length of the study was 14 weeks, of which the first two weeks were used to measure baseline physical activity (PA) to inform the input pseudo-random multisine signals. Twenty individuals participated in the study, and for each participant and idiographic model was studied using ARX modeling. The predictors (variables) measured in this study are in Table 3.1. The output variable of interest is measured in actual steps per day.

While models based on behavioral theory have been used to inform the decision rules which govern if and what kind of intervention suggestions should be sent, these theories have been insufficient to optimize the impact of the intervention. Further research into how the behavioral data should be used to inform the intervention is studied here by using Model-on-Demand as an empirical method to better understand an individual's walking behavior and predict the effectiveness of the interventions. These models could then be used to design control systems that can be further used to optimize the impact of these interventions.

3.3.2 Just Walk with MoD and Discrete SPSA

Applying discrete SPSA to Participant 073 of the *Just Walk* study, the following parameters were subjected to stochastic search: features (i.e. inputs), ARX orders (i.e. $n_a,\ n_b,\ n_k$) for each input/output, and local polynomial order (per Equation 2.4). The features vector consisted of 9 inputs: all 8 inputs from the *Just Walk* study and an interaction term between *expected points* and *predicted busyness*.

All features were initialized in a single vector, $\hat{\theta}^w \in \mathbb{R}^9$. The input vector is binary (0: the input is *not* used in the model, 1: the input is used in the model), and so the vector was bound between 0 and 1 and rounded such that it can only take on values of 0 or 1 when used to evaluate a model. Between iterations, the vector was allowed to take on values between 0 and 1, as it was updated using the estimated slope, but it was then rounded before being used to define the inputs being used for a specific model. The convergence of these values was not demonstrated as clearly as in cases with continuous-valued input vectors, as demonstrated in Schwartz *et al.* (2006).

The ARX orders ($n_a,\ n_b,\ n_k$) were each initialized as individual vectors, $\hat{\theta}^{n_a} \in \mathbb{R}^1$, $\hat{\theta}^{n_b} \in \mathbb{R}^9$, and $\hat{\theta}^{n_k} \in \mathbb{R}^9$. The values of $\hat{\theta}^{n_a}$ and $\hat{\theta}^{n_b}$ were both bound between 0 and 3, while $\hat{\theta}^{n_k}$ was bound between 0 and 1. These orders could take on higher values, but we chose to limit them to keep the regressor structure simple. The local polynomial order was also initialized as its own vector ($\hat{\theta}^P \in \mathbb{R}^1$), and was restricted between 0 and 2.

Similar to the input vector, the ARX orders and local polynomial order values were allowed take on any value between their respective bounds. However, they were rounded before evaluating the model.

The gain sequences were specified for each input vector (Table 3.2). These were determined by the relative size of approximated gradient compared to the values of the input vectors as well as the relative size of the perturbation. While these have to be determined by the user, they are typically not difficult to narrow down. The same c_k and a_k values were used for the input vectors corresponding to n_a, n_b, and n_k, but this does not have to be the case. These gains were sufficient to allow the search to span the full range of values for each of the parameters across the iterations. All values of c_k and a_k were kept constant between SPSA iterations, which is not the case for all SPSA implementations, but is used in binary applications of SPSA Aksakalli and Malekipirbazari (2016). The following fixed parameters were chosen for MoD: $k_{min} = 40$, $k_{max} = 400$, and the goodness of fit criterion was generalized cross validation (GCV).

Input Vector	c_k	a_k
θ^w	0.1	0.0002
θ^{n_a}, θ^{n_b}, θ^{n_k}	0.1	0.005
θ^P	0.2	0.004

Table 3.2: *Just Walk* MoD-DSPSA Gain Sequence Specification

As noted previously, the DSPSA algorithm is set up to maximize the weighted average of the MoD model's ability to fit (1) validation data, $J_v(\hat{\theta})$, (2) estimation data, $J_e(\hat{\theta})$, and (3) overall data, $J_o(\hat{\theta})$, where $J_{\{v,e,o\}}$ corresponds to the Normalized Root-Mean-Square Error (NRMSE), which is calculated at each iteration by Eqn. 3.3

$$J_{\{v,e,o\}}(\theta) = 100\% \times \left(1 - \frac{||y - \hat{y}||_2}{||y - \bar{y}||_2}\right) \tag{3.3}$$

in which $\hat{y}$ is the model output, while $\bar{y}$ is the average of the data, and $||\cdot||_2$ is the 2-norm. The fit percentages were then weighted as 4/6, 1/6, and 1/6, respectively. The predictive fit was weighted heavier than both the estimation and overall fits, since the predictive fit is typically much lower than the other two (and sometimes negative) and since the predictive ability of the model serves a significant purpose in future behavioral health interventions, namely to predict whether a participant is in a state or environment conducive towards their behavioral goals (i.e. walking more). These weights and fits then give us our objective function, which in this case takes on the form of a maximization problem (or minimization of the negative of the weighted fit):

$$\max_{\theta \in \mathbb{Z}_+} J(\theta) \tag{3.4}$$

$$J(\theta) = \frac{4}{6} J_v(\theta) + \frac{1}{6} J_e(\theta) + \frac{1}{6} J_o(\theta) \tag{3.5}$$

in which $\theta^w,\ \theta^{n_a},\ \theta^{n_b},\ \theta^{n_k},\ \theta^P \in \theta$, and the optimized solution found by SPSA is θ^*. As shown by Eqn. 3.4 and 3.5, all input vectors are used to simultaneously update the same objective function. They are not evaluated independently.

3.3.3 Just Walk Application Results

The NRMSE fit for both MoD and ARX model's evaluated at the k^{th} iteration of inputs (features, ARX orders, local polynomial order) are shown in Figure 3.1. The largest weighted average of the MoD fits occurs at $k = 22$. The exact fits of the model evaluated at the optimal iteration are listed in Table 3.3. In each evaluation, the fit provided by the MoD model is just under double that of the ARX model. The model evaluations on each data set outlined in Table 3.3 are shown in further detail in Figures 3.2, and 3.3.

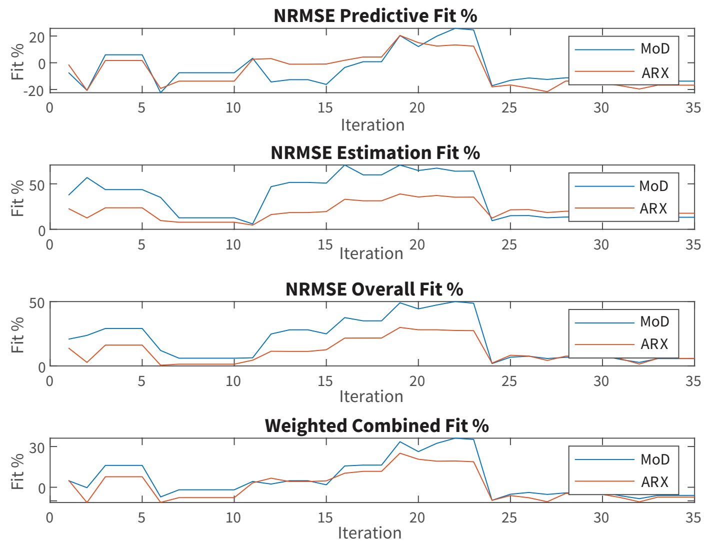

Figure 3.1: *Just Walk*: NRMSE Fit (%) Per DSPSA Iteration

	NRMSE Fit (%)		RMS Error (steps)		Max Error (steps)	
	MoD	ARX	MoD	ARX	MoD	ARX
Prediction	25.73	13.35	1877.42	2192.41	5585.31	4289.08
Estimation	63.89	35.52	837.01	1499.33	3876.73	4208.57
Overall	50.03	27.55	1189.39	1724.26	5536.66	4807.34

Table 3.3: *Just Walk*: MoD and ARX Comparisons for θ^*

The features used in the optimal iteration as well as their respective n_b and n_k orders are outlined in Table 3.4. These are also the features used in the model evaluated in Figures 3.2 and 3.3. The n_a value is common to all inputs, so there is only one value of n_a being optimized. In this case, the optimal n_a value found by DSPSA is $n_a = 2$, which means that two prior lags of the output (steps) are being used in the model.

In Figures 3.2 and 3.3, the validation data is taken from the last 25% of the

Initialized Features	Selected Features	n_b	n_k
Goals	Goals	1	1
Expected Points	Expected Points	2	0
Granted Points	Granted Points	2	0
Predicted Busyness	Predicted Busyness	1	0
Predicted Stress	Predicted Stress	0	1
Predicted Typical	Predicted Typical	2	1
Weekend	Weekend	1	1
Temperature	-	-	-
Expected Points/Predicted Busyness	-	-	-

Table 3.4: *Just Walk*: MoD-DSPSA Feature Selection and n_b, n_k orders for θ^* ($n_a = 2$, $P = 1$)

participant's data, while the former 75% is used to estimate the model. The changes in neighborhood size, used by MoD to estimate a local model, is also shown in the middle plot of both figures.

Figure 3.4 illustrates the features used in both the MoD and ARX models at each iteration. The features used in the optimal iteration and demonstrated in Figures 3.2 and 3.3 are: *goals, expected points, granted points, predicted busyness, predicted stress* and *weekend*, which is a reduced set of features from the original nine.

The local polynomial order used in each iteration is shown in Figure 3.5. The bottom plot shows the exact value of the polynomial order after being updated by the gradient approximation after each iteration. These values are then rounded to be used in each model iteration. The rounded values are shown in the upper plot. The local polynomial order is only used in the MoD models.

Figure 3.2: *Just Walk*: MoD and ARX Model Comparison on Validation Data (25% of Participant Data). Output Measured in Steps.

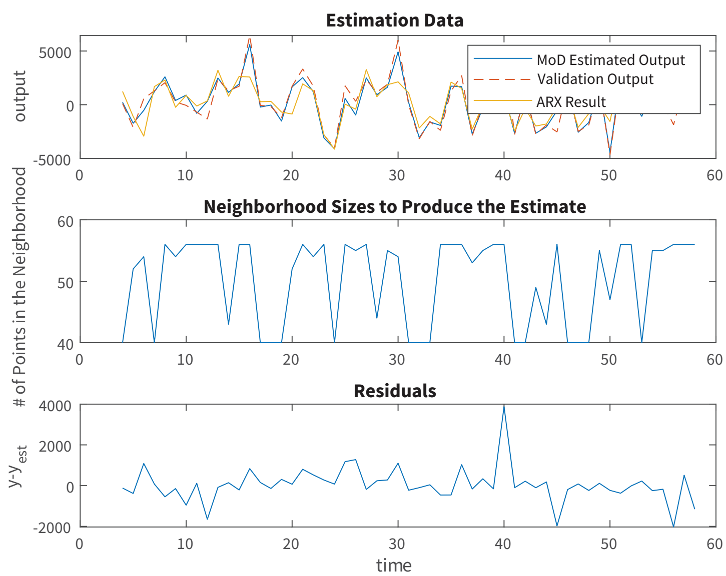

Figure 3.3: *Just Walk*: MoD and ARX Models Comparison on Estimation Data (75% of Participant Data). Output Measured in Steps.

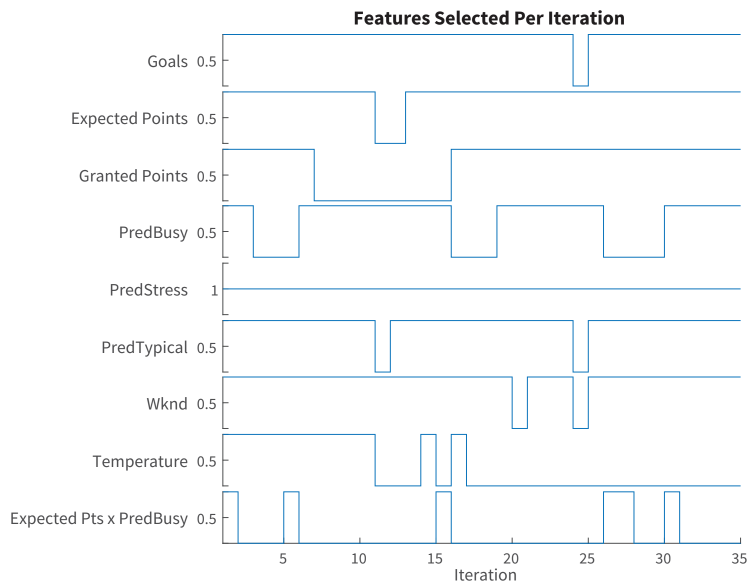

Figure 3.4: *Just Walk*: MoD-DSPSA Features Used at Each Iteration ($\hat{\theta}_k^w$)

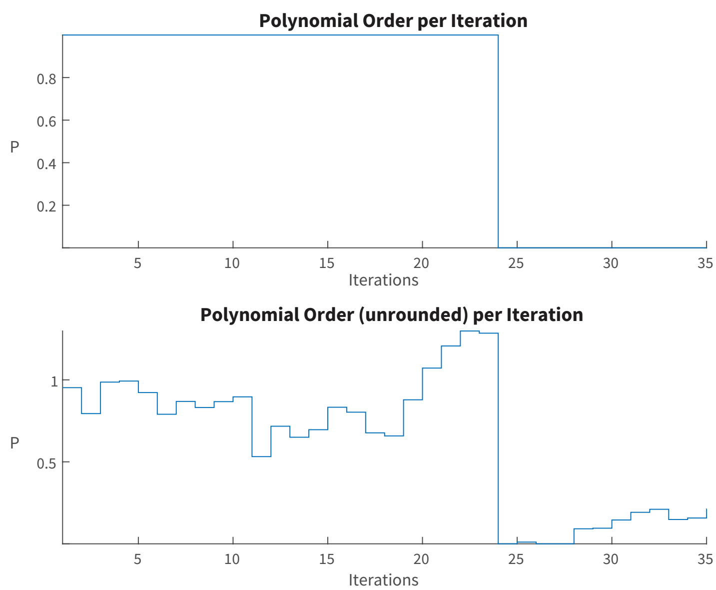

Figure 3.5: *Just Walk*: MoD-DSPSA Local Polynomial Order at Each Iteration ($\hat{\theta}_k^P$)

The n_a, n_b, and n_k values evaluated at each iteration, shown in Figures 3.6, 3.7, and 3.8, are updated for all initialized inputs, regardless of which inputs are used in

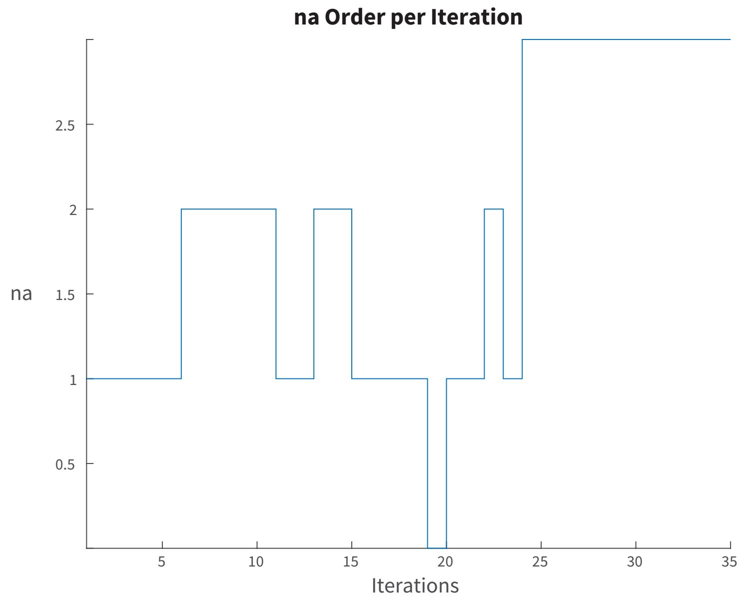

Figure 3.6: *Just Walk*: MoD-DSPSA n_a ($\hat{\theta}_k^{n_a}$) Values at Each Iteration (Rounded and Bound Between 0 and 3)

the model at that same iteration. However, the ARX orders are only used in the model if its corresponding feature is also used in the model (which can be identified in Figure 3.4). An additional constraint was placed on the DSPSA search regarding n_b and n_k values, since they cannot simultaneously be zero in the regressor structure. So, n_k values were set to 1 whenever n_b values were 0.

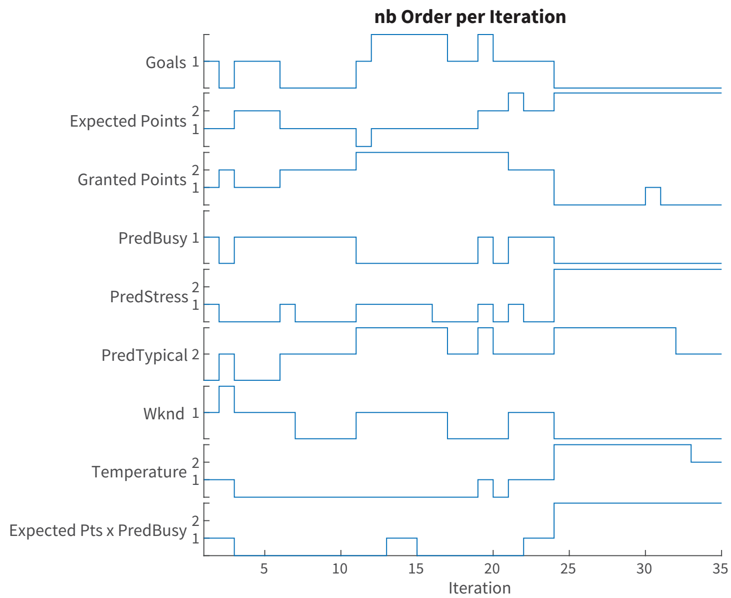

Figure 3.7: *Just Walk*: MoD-DSPSA n_b ($\hat{\theta}_k^{n_b}$) Values at Each Iteration (Rounded and Bound Between 0 and 3)

3.4 Summary and Conclusions

The results obtained from the *Just Walk* case study show that Model-on-Demand estimation can provide better individualized models than ARX, with both more explanative and predictive power. Though the challenge of choosing the right parameters to estimate the MoD model can be tedious and require a search over many combinations, DSPSA provides an efficient means to determine the inputs of the highest-performing models. DSPSA can be used not only for feature selection but can also provide insights into additional model parameters such as the MoD local polynomial order and ARX orders. DSPSA also bridges the gap between using MoD and using ARX in terms of modeling expertise required to tune MoD's adjustable parameters. The case shown here only searched over regressor orders and the local polynomial order, but DSPSA can readily be used to search over other MoD parameters (i.e. goodness of fit

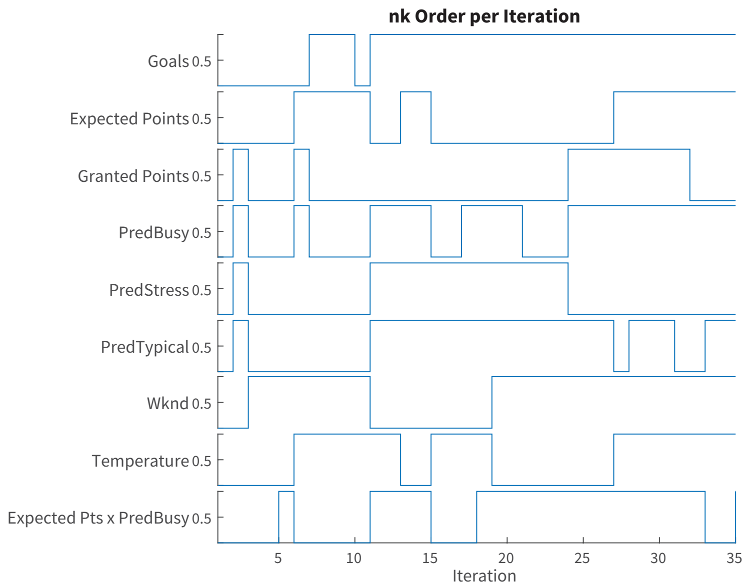

Figure 3.8: *Just Walk*: MoD-DSPSA n_k ($\hat{\theta}_k^{n_k}$) Values at Each Iteration (Rounded and Bound Between 0 and 1)

criterion). With DSPSA as a tool to quickly define and evaluate MoD models, these models can then be used to design controllers such as in the form of model predictive control (MPC), as shown in Nandola and Rivera (2010).

While the parameters during DSPSA do not appear to clearly converge as it performs 'gradient descent,' DSPSA still allows users to obtain an optimized set of model parameters efficiently and with modest effort, while avoiding a brute force search. However, this highlights one limitation in SPSA, in that there is no 'natural stopping point' as the algorithm runs until it hits a stall limit or finishes the iterations defined by the user (common to both continuous and discrete applications). With DSPSA, the inability to observe a clear convergence also prevents users from readily inferring whether the model identified is *the* best model (or the near best model). Due to the discrete inputs (and lack of an explicit function), there is also already a limit to our ability to infer how combinations of features contribute to model proficiency (i.e. whether taking one feature away or adding one will help or hurt the model, given the other features already being implemented); this problem is not unique to the SPSA framework. Visualization of the exact iterations or parameter values in continuous space may still be useful for analyses. These may allow DSPSA users to observe the step size taken at each iteration, to better understand the approximated gradient resulting from the particular simultaneous perturbation.

Despite these limitations, using DSPSA in conjunction with MoD has demonstrated that MoD can provide better models than ARX for behavioral health interventions. This also highlights the need for more robust modeling for individualized solutions, which is better achieved by MoD than ARX. Given the ease by which DSPSA can be set up and the efficiency by which it finds near-optimal models, using DSPSA also provides a method to find idiographic dynamic models for personalized interventions for a large number of participants, making the concept of personalized interventions scalable.

Though this chapter demonstrates the effectiveness of MoD coupled with DSPSA, the next chapter will address the choice of estimation and validation data from a given individual's walking data. The results demonstrated here partitioned the participant's 80-days of data by reserving the first 75% for estimation data and the latter 25% for validation data, which (1) may not be the optimal choice to maximize the predictive and explanatory power of the MoD models and (2) may introduce other sources of bias into the models, including those related to the progress of the intervention itself (i.e. notification fatigue, habituation, etc.). In the next chapter, DSPSA will be used with ARX modeling to study how segmenting an individual's data into estimation and validation sets can influence model development.

Chapter 4

ARX ESTIMATION WITH DSPSA

4.1 Motivations

Chapter 3 demonstrated that Model-on-Demand (MoD) models optimized with DSPSA can efficiently provide improved models of walking behavior. While DSPSA was used to search over combinations of various parameter values and features, an important issue in developing behavioral models is the choice of estimation and validation data. As previously explained, behavioral interventions result in one data set per individual, rather than multiple data sets that can be used for estimation and validation data as was done in the CSTR case studies in Chapter 2. In the results shown in Chapter 3, the individual's data was split 75%/25%, in which the first 75% of the data was reserved as estimation data for MoD and the latter was used as validation data. However, this may not be the best way to split the data. Some reasons include notification fatigue, habituation, or general influences related to time or the progression of the study, which can introduce bias into the model. There also little to no *a priori* justification for any other partitioning (i.e. 50%/50%, validation first/estimation last, etc.).

The *Just Walk* study used repeating cycles of a pseudo-random multisine signal for each of the two input features, *Goals* and *Expected Points*. Each cycle was 16 days long, which was repeated five times to provide the full intervention data set. So, rather than using the last 25% of the individual's data as validation data and the former 75% as estimation data, it is possible to mix and match data from each of the five cycles to form the estimation and validation data sets. A similar strategy was done in a previous study by Freigoun *et al.* (2017), which assessed various combinations of the five cycles as estimation and validation data on the model fit achieved by ARX estimation. Only combinations of two or three cycles for each estimation or validation were considered, for a total of twenty possible cycle combinations. The optimal ARX order for each combination was determined through exhaustive search with fixed inputs, in a procedure that took days and relied on high performance computers. In this chapter, the procedure was repeated but instead with DSPSA to find the optimal ARX orders and features as a more efficient and less computationally-demanding alternative.

4.2 *Just Walk*: Participant Walking Data

In the following sections, data from select participants will be used to demonstrate the combination of DSPSA as a search technique with various combinations of estimation and validation data. The participant shown for the results demonstrated in Freigoun *et al.* (2017), Participant 230, is known as the 'operant learning' participant, as they were particularly attentive to the goals set by the intervention resulting in behavior that was especially favorable for modeling. This is not the case with every *Just Walk* participant, as demonstrated in following sections.

The input-output data for Participant 230 is shown in Figure 4.1. This data is segmented into five sections corresponding to the five units of the repeated input signal (multi-sinusoidal signals for goals and expected points).

Walking data from two additional participants, Participant 008 and Participant 057, will also be used to demonstrate DSPSA as an optimization technique to find ARX orders and features that best explain and predict their respective behavior. The input-output data for Participant 008 is shown in Figure 4.2, and the input-output data for Participant 057 is shown in Figure 4.3. Note that while the first four cycles of data are each 16 days, the fifth cycle is slightly longer for all three participants. All data shown is mean subtracted.

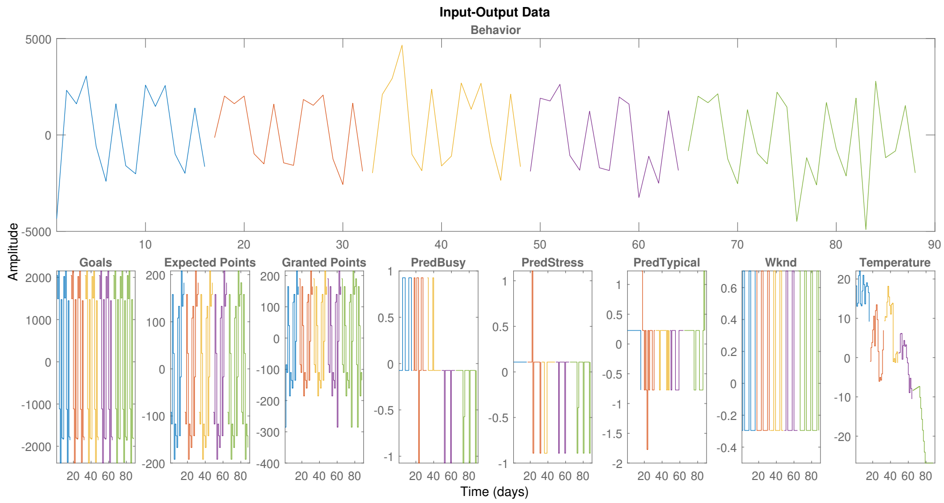

Figure 4.1: *Just Walk*: Participant 230 Mean-Subtracted Input-Output Data Segmented into Five Cycles

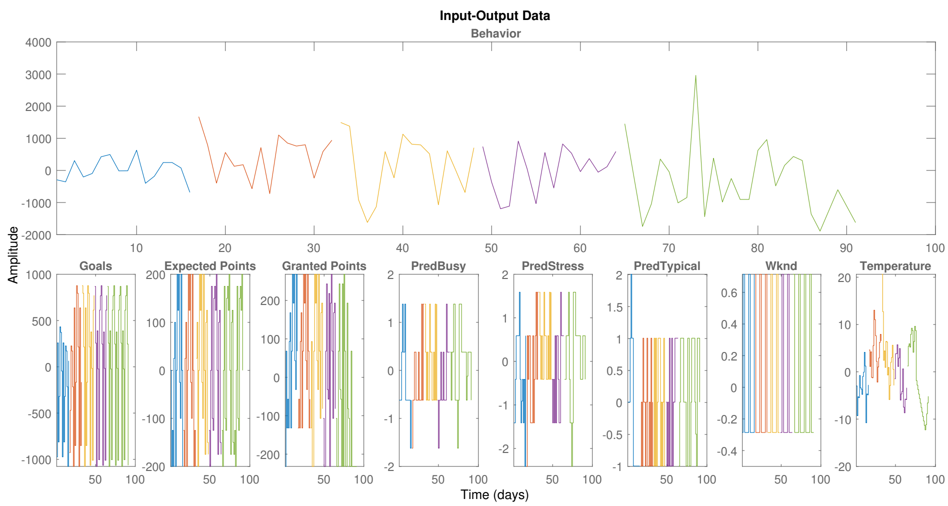

Figure 4.2: *Just Walk*: Participant 008 Mean-Subtracted Input-Output Data Segmented into Five Cycles

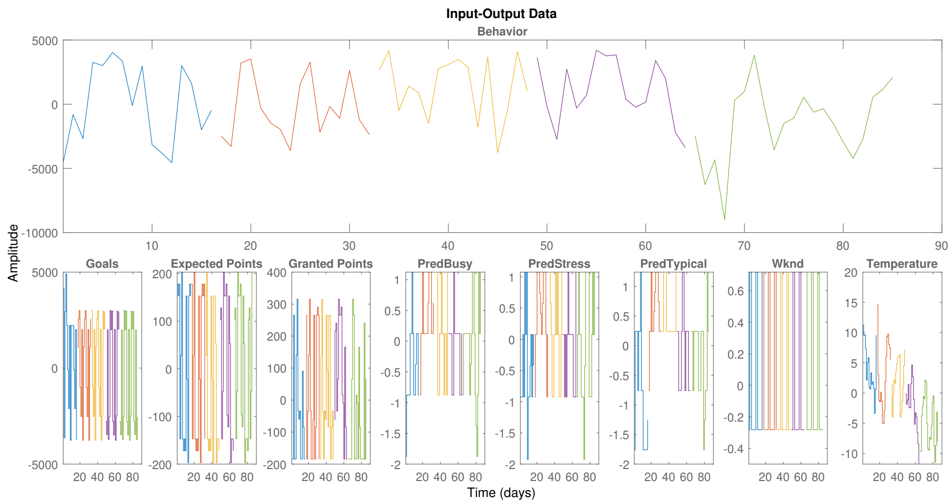

Figure 4.3: *Just Walk*: Participant 057 Mean-Subtracted Input-Output Data Segmented into Five Cycles

4.3 *Just Walk* Participant 230: 4-Input ARX Models with DSPSA

In this section, discrete SPSA is used to optimize the regressor orders for ARX estimation models of walking behavior using various combinations of estimation and validation data from the *Just Walk* study. This is be compared with the results from Freigoun *et al.* (2017), which used exhaustive search to obtain the optimal ARX orders; these results are shown in Figure 4.4.

The results in Figure 4.4 were computed for a 4-input model for a select participant. The four inputs are: *Goals, Expected Points, Granted Points,* and *Predicted Busyness*. Only n_a and n_b values were searched over, and n_k for each input was kept constant at $n_k = 1$. Bounds were placed on both n_a and n_b values, such that neither could take on values less than 1 or greater than 3. These search conditions were also used for the SPSA search.

Unlike the previous chapter, only n_a and n_b, were be optimized by SPSA. These were initialized as separate input vectors, and the gains for each are listed in Table 4.1.

E*	V*	NRMSE Fit (%)					Average Estimation NRMSE Fit (%)	Average NRMSE Validation Fit (%)	Overall NRMSE Fit (%)	ARX Order (4-input) $[na_1,nb_1,nb_2,nb_3,nb_4]$
		Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5				
[1,2]	[3,4,5]	77.40%	85.44%	79.27%	27.68%	13.70%	81.42%	40.22%	40.11%	[1,1,1,1,3]
[1,3]	[2,4,5]	77.39%	82.25%	81.30%	26.88%	15.36%	79.35%	41.50%	40.60%	[1,2,1,1,3]
[1,4]	[2,3,5]	64.82%	71.25%	67.27%	45.89%	21.04%	55.36%	53.19%	42.29%	[1,3,1,1,1]
[1,5]	[2,3,4]	61.36%	59.51%	60.96%	40.14%	24.47%	42.92%	53.54%	37.40%	[1,1,1,3,1]
[2,3]	[1,4,5]	70.46%	90.25%	84.15%	25.00%	11.19%	87.20%	35.55%	37.70%	[3,3,1,2,3]
[2,4]	[1,3,5]	49.06%	71.94%	67.25%	52.39%	22.98%	62.17%	46.43%	40.56%	[3,1,2,1,3]
[2,5]	[1,3,4]	54.89%	61.75%	60.36%	47.35%	23.68%	42.72%	54.20%	39.33%	[3,1,1,1,1]
[3,4]	[1,2,5]	45.97%	61.27%	69.24%	51.46%	24.02%	60.35%	43.75%	41.15%	[1,3,3,1,3]
[3,5]	[1,2,4]	63.11%	66.96%	52.29%	41.52%	19.47%	35.88%	57.20%	41.12%	[1,1,1,1,1]
[4,5]	[1,2,3]	36.37%	52.47%	50.06%	49.24%	25.88%	37.56%	46.30%	32.75%	[1,1,1,3,2]
[3,4,5]	[1,2]	53.63%	64.61%	49.26%	46.59%	19.93%	38.59%	59.12%	40.12%	[1,1,1,1,1]
[2,4,5]	[1,3]	50.12%	59.76%	59.36%	49.92%	23.64%	44.44%	54.74%	38.71%	[3,1,1,1,1]
[2,3,5]	[1,4]	58.63%	66.76%	64.91%	49.62%	27.28%	52.98%	54.13%	40.59%	[3,1,3,2,1]
[2,3,4]	[1,5]	59.43%	76.99%	70.11%	41.51%	22.32%	62.87%	40.88%	41.61%	[2,3,3,2,3]
[1,4,5]	[2,3]	57.91%	61.11%	60.18%	45.69%	24.92%	42.84%	60.65%	38.81%	[1,1,1,3,1]
[1,3,5]	[2,4]	66.34%	66.02%	67.24%	42.13%	22.57%	52.05%	54.08%	41.31%	[1,3,1,1,1]
[1,3,4]	[2,5]	68.39%	77.75%	73.46%	41.86%	18.78%	61.24%	48.27%	42.26%	[1,3,2,1,1]
[1,2,5]	[3,4]	61.85%	56.05%	68.43%	44.82%	35.02%	50.97%	56.63%	46.03%	[2,3,1,2,3]
[1,2,4]	[3,5]	71.99%	73.18%	72.36%	43.28%	20.40%	62.82%	46.38%	43.61%	[1,2,1,1,3]
[1,2,3]	[4,5]	75.95%	87.02%	80.67%	26.39%	13.36%	81.21%	19.88%	39.87%	[1,1,1,1,3]

*E≡Estimation Cycles (magenta), V≡Validation Cycles (cyan)

Figure 4.4: *Just Walk:* Participant 230 4-Input ARX Model Exhaustive Search Results (Reprinted from Freigoun *et al.* (2017))

Input Vector	c_k	a_k
θ^{n_a}	0.1	0.005
θ^{n_b}	0.1	0.005

Table 4.1: *Just Walk*: ARX-DSPSA Gain Sequence Specification for n_a, n_b

To implement SPSA in this context, separate input vectors were initialized for each n_a and n_b. The n_a vector has length equal to the number of outputs, while the vector for n_b each has length equal to the number of outputs. The SPSA algorithm was then run for each set of estimation and validation data.

Using SPSA, 70 iterations were run for each set of estimation and validation cycle. The total time for 70 iterations to be completed for all 20 possible sets was approximately 9.2 minutes. These were computed on a Lenovo Yoga ThinkPad with

			NRMSE Fit (%)					Avg Est	Avg Val	Avg Overall	Obj Fxn	ARX Order (4-input)
	E*	V*	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	%	%	%	%	[na,nb1,nb2,nb3,nb4]
1	[1,2]	[3,4,5]	85.00	82.54	69.21	32.35	14.31	83.77	38.62	41.80	35.70	[1,2,1,1,3]
2	[1,3]	[2,4,5]	84.81	66.30	70.88	23.11	8.89	77.85	32.77	36.47	26.91	[1,1,1,1,3]
3	[1,4]	[2,3,5]	85.19	73.92	71.70	48.01	33.85	66.60	59.82	49.63	48.61	[3,3,3,2,1]
4	[1,5]	[2,3,4]	69.82	64.85	51.38	40.17	20.98	45.40	52.13	41.74	45.98	[2,1,1,1,2]
5	[2,3]	[1,4,5]	83.33	78.80	70.80	25.90	11.50	74.80	40.25	39.15	33.09	[1,1,1,1,3]
6	[2,4]	[1,3,5]	62.82	70.67	61.77	50.28	20.70	60.48	48.43	43.01	40.72	[1,2,1,1,3]
7	[2,5]	[1,3,4]	63.17	65.45	47.64	42.98	21.02	43.23	51.26	41.08	46.35	[1,1,1,1,1]
8	[3,4]	[1,2,5]	58.09	52.82	69.25	50.54	17.10	59.90	42.67	39.67	35.79	[2,3,1,1,2]
9	[3,5]	[1,2,4]	71.43	66.42	53.64	42.66	19.65	36.64	60.17	42.74	51.77	[1,1,1,1,1]
10	[4,5]	[1,2,3]	54.15	55.54	43.74	46.83	21.86	34.35	51.14	38.98	45.07	[1,1,1,1,1]
11	[3,4,5]	[1,2]	66.41	68.87	52.09	47.49	20.14	39.91	67.64	42.40	55.67	[1,1,1,2,1]
12	[2,4,5]	[1,3]	61.47	64.94	48.64	46.31	20.46	43.91	55.06	41.39	48.79	[1,1,1,1,1]
13	[2,3,5]	[1,4]	76.51	76.67	60.13	42.51	19.85	52.22	59.51	43.16	51.44	[1,1,1,3,1]
14	[2,3,4]	[1,5]	72.27	73.73	66.70	45.25	17.53	61.89	44.90	42.97	37.33	[2,3,1,1,1]
15	[1,4,5]	[2,3]	68.36	68.63	51.44	45.44	20.86	44.89	60.04	42.61	50.62	[2,1,1,1,1]
16	[1,3,5]	[2,4]	80.68	74.32	62.19	39.75	19.14	54.00	57.03	42.72	47.29	[1,1,1,3,1]
17	[1,3,4]	[2,5]	80.42	70.44	70.69	39.91	14.51	63.67	42.48	41.75	31.96	[2,3,1,1,1]
18	[1,2,5]	[3,4]	68.45	57.73	66.93	40.51	31.80	52.66	53.72	46.93	50.62	[1,1,1,2,3]
19	[1,2,4]	[3,5]	76.46	78.23	57.91	41.86	16.40	65.52	37.15	42.82	35.44	[1,2,1,1,1]
20	[1,2,3]	[4,5]	85.00	78.83	70.58	28.48	13.73	78.14	21.10	41.01	27.14	[1,1,1,1,3]

Figure 4.5: *Just Walk*: Participant 230 4-Input ARX Model Results found by DSPSA The models with the highest SPSA objective function values are highlighted in yellow.

an Intel Core i7-10510U processor. The objective function to assess each set of ARX parameters was a weighted average of the ARX NRMSE fit to the validation data (60%) and and overall data (40%):

$$J(\theta) = 0.6J_v(\theta) + 0.4J_o(\theta) \tag{4.1}$$

in which each NRMSE fit term (J_v, J_o) is calculated by Equation 4.2, where y corresponds to the data from each respective partition of data (i.e. validation or overall, respectively).

$$J_{\{v,o\}}(\theta) = 100\% \times \left(1 - \frac{||y - \hat{y}||_2}{||y - \bar{y}||_2}\right) \tag{4.2}$$

Equation 4.1 is different from the criterion used in Chapter 3, in which a weighted average of all three fits to validation, estimation, and overall data was used. Here, only validation and overall fits (with higher weight placed on the validation fit) were used to more closely match the process used in Freigoun *et al.* (2017) to determine the best models. In Freigoun *et al.* (2017), the researchers first looked at the models with the highest validation fits, and then from that set focused on those which had the highest overall fits. The overall fit accounts for the estimation data, which does not need to be considered separately.

The results of applying DSPSA to search over n_a and n_b of the four fixed inputs are summarized in Figure 4.5. Comparing the "best" models found by DSPSA with the results found by exhaustive search in Figure 4.4, DSPSA was able to obtain similar NRMSE fit percentages for each set of estimation and validation data, but in much less time.

From Freigoun *et al.* (2017), the set of estimation and validation data that led to the best model was determined to be set #18, which used three cycles of data for estimation (cycles 1,2, and 5) and two cycles of data for validation (cycles 3 and 4). The results of both methods (exhaustive search and discrete SPSA) are shown in Table 4.2, and the DSPSA results are comparable to those found by exhaustive search.

Method	Avg. Est. NRMSE %	Avg. Val. NRMSE %	Avg. Overall NRMSE %	ARX Order $[n_a, n_{b,1}, n_{b,2}, n_{b,3}, n_{b,4}]$
Exhaustive Search	50.97	56.63	46.03	[2,3,1,2,3]
DSPSA	52.66	53.72	46.93	[1,1,1,2,3]

Table 4.2: *Just Walk*: 4-Input ARX Model Comparison for Est/Val Set # 18, with Estimation Cycles 1, 2, and 5; Validation Cycles 3 and 4

The best model for the estimation/validation set #18 was found at the 14th iteration of SPSA (out of 70 iterations for the given set). This is shown in Figure 4.6. As mentioned in the table in Figure 4.5, the ARX orders found by DSPSA are $n_a = 1$, meaning that only one previous lag of the output of interest (step count) was used. The n_b orders for each feature (*Goals*, *Expected Points*, *Granted Points*, and *Predicted Busyness*) were 1, 1, 2, 3, respectively; for *Goals* and *Expected Points* only one prior input was needed. For *Granted Points* and *Predicted Busyness*, two and three prior inputs were used, respectively.

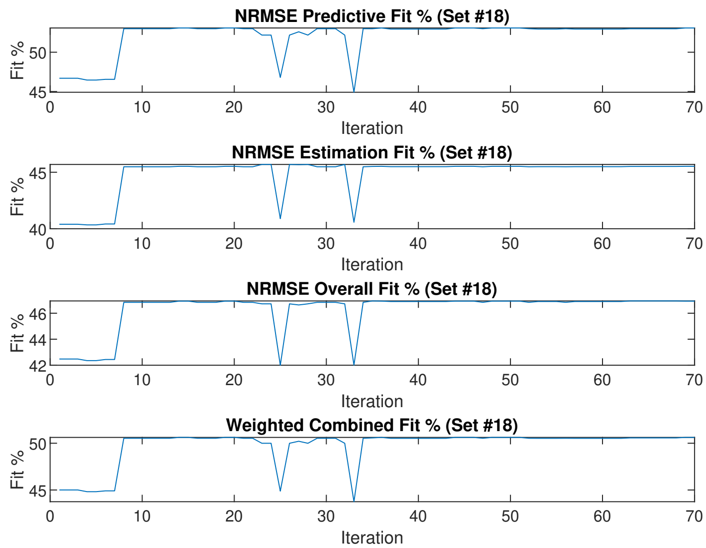

Figure 4.6: Just Walk: Participant 230 4-Input ARX Model Objective Function DSPSA Iterations, Est/Val Set #18

This model on the overall data (estimation and validation combined) is shown in Figure 4.7. The NRMSE fit % of the model is 46.9%. This model applied to the estimation and validation data sets is shown in Figure 4.8, in which the fit to the estimation data is 52.6% and the fit to the validation data is 53.7%. Since only n_a and n_b were subjected to DSPSA, the values used at each iteration are shown in Figures 4.9 and 4.10. The values at iteration 14 of the DSPSA search are the ones used in the best model found, as noted in Figure 4.5.

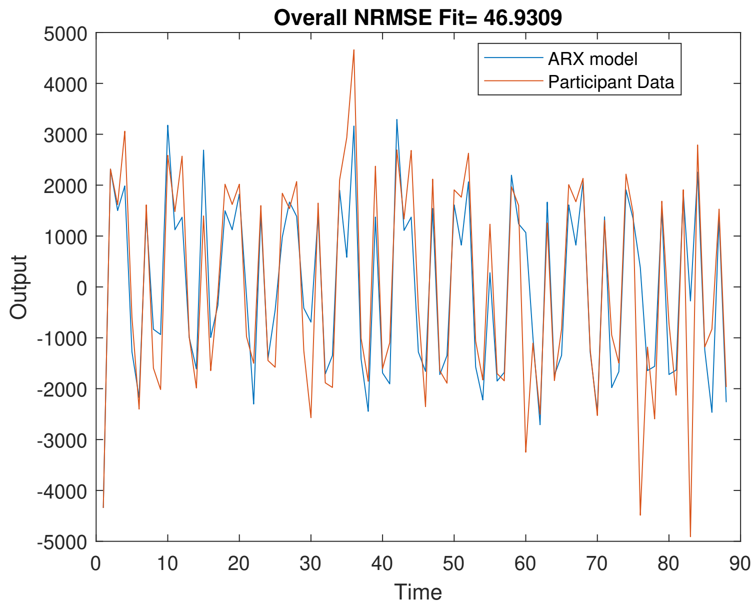

Figure 4.7: *Just Walk*: Participant 230 4-Input ARX Model on Overall Data #18

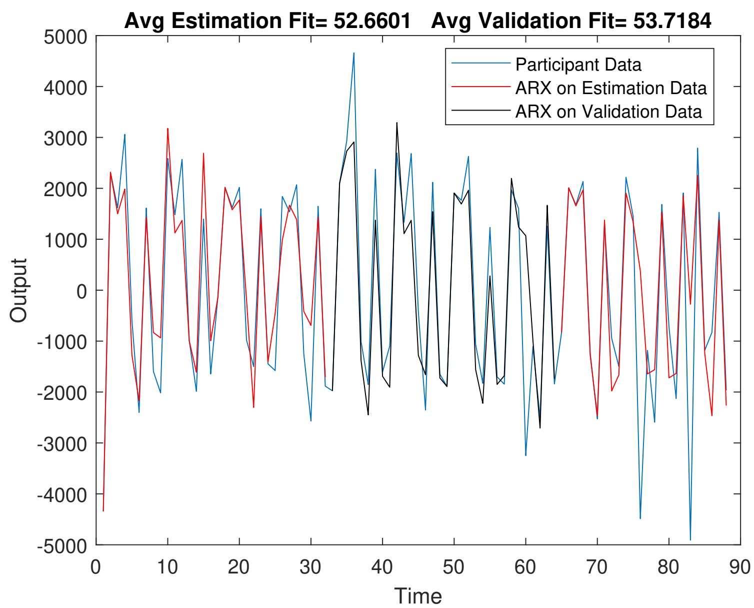

Figure 4.8: *Just Walk*: Participant 230 4-Input ARX Model on Estimation and Validation Data for Set #18

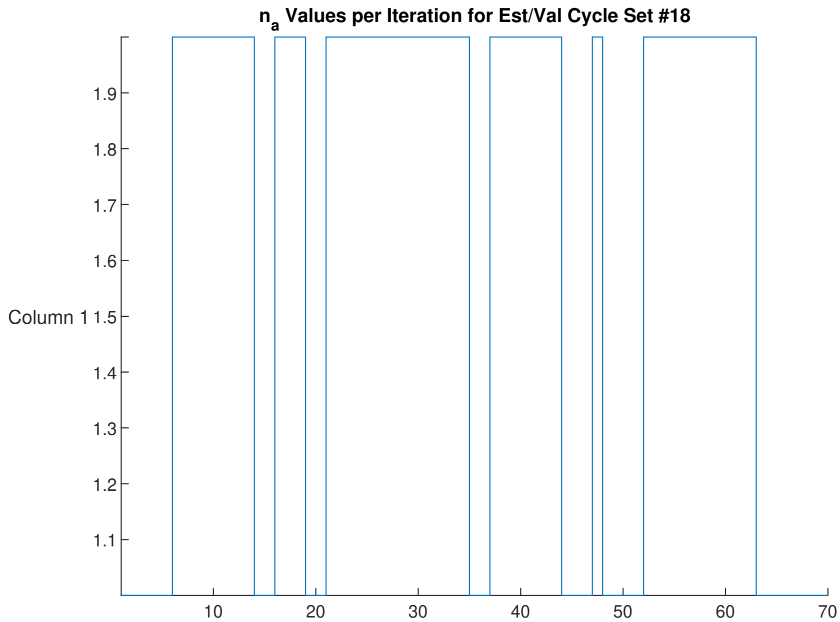

Figure 4.9: *Just Walk*: Participant 230 n_a Iterations for 4-Input ARX Model, Set #18

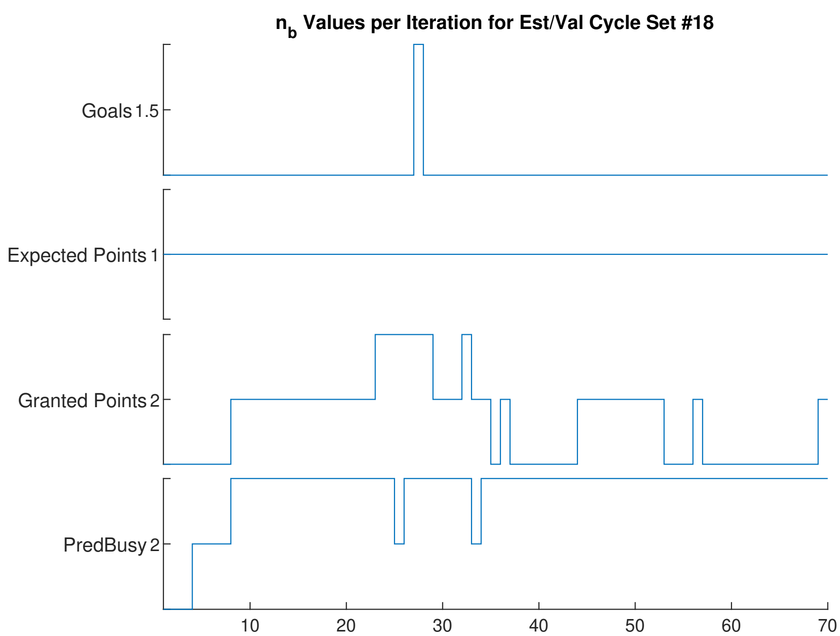

Figure 4.10: *Just Walk*: Participant 230 n_b Iterations for 4-Input ARX Model, Set #18

4.4 *Just Walk* Participant 230: Feature Selection with n_a, n_b, and n_k Search in ARX Model Estimation with DSPSA

In this section, Participant 230 is again examined as in Section 4.3. Discrete simultaneous perturbation stochastic approximation (DSPSA) was used to search over both features and all three ARX orders n_a, n_b, and n_k. Previously, both the input features and n_k values were fixed. For feature selection, all 8 inputs were included in the search. Here, n_k is allowed to take on values of either 0 or 1. An n_k value of zero denotes that there is no delay, that is, the impact of changes to a respective feature occur at the same sampling time; whereas an n_k value of 1 denotes that there is a delay of 1, meaning that the impact develops one sampling time later. In this application, the sampling time is 1 day, and so $n_k = 0$ refers to a within-day effect, while $n_k = 1$ refers to a day after effect.

			NRMSE Fit (%)					Avg Est	Avg Val	Avg Overal	Obj Fxn
	E*	V*	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	%	%	%	%
1	[1,2]	[3,4,5]	87.16	82.51	70.73	79.80	69.04	84.84	73.19	72.80	71.86
2	[1,3]	[2,4,5]	86.99	68.78	75.00	72.34	65.98	80.99	69.04	70.58	68.54
3	[1,4]	[2,3,5]	89.79	80.44	72.22	88.66	76.61	89.23	76.42	76.67	74.78
4	[1,5]	[2,3,4]	88.49	78.44	69.44	83.85	78.16	83.32	77.24	76.74	75.67
5	[2,3]	[1,4,5]	85.68	77.98	71.14	78.13	74.09	74.56	79.30	74.34	76.10
6	[2,4]	[1,3,5]	86.15	84.41	74.08	89.25	78.04	86.83	79.42	76.52	75.91
7	[2,5]	[1,3,4]	83.43	82.79	66.58	84.45	80.28	81.54	78.15	76.54	75.62
8	[3,4]	[1,2,5]	87.66	73.54	77.64	85.50	72.80	81.57	78.00	74.97	74.25
9	[3,5]	[1,2,4]	88.57	74.14	73.69	83.12	76.47	75.08	81.94	76.38	77.79
10	[4,5]	[1,2,3]	85.96	81.81	70.86	84.91	80.61	82.76	79.54	77.13	75.46
11	[3,4,5]	[1,2]	88.66	79.45	74.45	86.14	78.53	79.71	84.06	77.50	79.00
12	[2,4,5]	[1,3]	85.22	82.95	70.88	85.15	81.08	83.06	78.05	77.31	75.76
13	[2,3,5]	[1,4]	87.55	79.25	71.55	84.26	77.60	76.13	85.91	76.77	82.16
14	[2,3,4]	[1,5]	86.92	76.87	79.54	87.67	75.98	81.36	81.45	76.04	77.07
15	[1,4,5]	[2,3]	87.86	79.07	73.33	85.99	80.72	84.86	76.20	77.87	74.94
16	[1,3,5]	[2,4]	88.87	78.23	73.92	84.62	78.98	80.59	81.43	77.20	79.36
17	[1,3,4]	[2,5]	88.13	76.36	71.40	85.19	74.34	81.57	75.35	74.92	74.02
18	[1,2,5]	[3,4]	85.70	82.10	70.58	81.60	81.28	83.02	76.09	76.85	75.00
19	[1,2,4]	[3,5]	87.89	81.61	73.02	88.21	76.49	85.91	74.75	76.75	75.49
20	[1,2,3]	[4,5]	88.73	74.27	74.58	81.40	75.78	79.19	78.59	76.45	76.69

Figure 4.11: *Just Walk*: Participant 230 Results for ARX Models found by Discrete SPSA with Feature Selection and n_a, n_b and n_k Search (Best cases highlighted in yellow)

For this search, the number of DSPSA iterations was increased from 70 to 110 per set of estimation and validation data. The total time for search was 13.6 minutes. Allowing for search of n_k led to much higher NRMSE fit percentages on average, than either of the prior searches (4-input fixed model, feature selection). The results for each set of data are tabulated in Figure 4.11. The three models with the largest objective function value (as per Equation 4.1) are highlighted in yellow. These are the models for data sets #11, #13, and #16.

The features used in each of the best models (for the 20 sets of estimation and validation data) are shown in Figure 4.12, along with their corresponding n_a, n_b and n_k values. The most commonly used features were *Goals, Expected Points, Granted Points*, and *Predicted Typical*, which were used in all 20 models.

		FEATURES								ARX		
	Total Inp	Goals	Exp Pts	Granted Pts	PredBusy	PredStress	PredTyp	Weekend	Temp	na	nb	nk
1	4	x	x	x		x				3	3 1 1 1	0 1 0 1
2	4	x	x	x		x				2	2 1 1 1	0 1 0 1
3	5	x	x	x		x		x		1	3 1 1 1 3	1 1 0 0 0
4	3	x	x	x						2	2 2 2	1 1 0
5	4	x	x	x			x			1	1 1 1 1	1 1 0 1
6	5	x	x	x			x			3	3 3 3 1	1 1 0 1
7	4	x	x	x			x			1	1 1 3 3	1 1 0 0
8	3	x	x	x						3	3 1 1	0 1 0
9	3	x	x	x						2	2 2 2	0 1 0
10	3	x	x	x						3	1 1 1	1 1 0
11	3	x	x	x						3	3 1 1	0 1 0
12	3	x	x	x						3	1 3 1	1 1 0
13	3	x	x	x						2	2 2 1	1 1 0
14	3	x	x	x						3	3 3 3	1 0 0
15	3	x	x	x						3	1 3 1	1 1 0
16	3	x	x	x						3	1 1 1	1 1 0
17	3	x	x	x						1	3 1 1	0 1 0
18	3	x	x	x						3	3 3 1	0 1 0
19	3	x	x	x						3	3 3 3	0 1 0
20	3	x	x	x						2	2 3 3	0 1 0

Figure 4.12: *Just Walk*: Participant 230 Feature Selection for Each Set of Estimation and Validation Data with Corresponding n_a, n_b and n_k Values

To compare with the prior section, the DSPSA search and best model found for set #18 can be found in Appendix C. However, it is important to note that based on the objective function defined in Equation 4.1, set #18 was not found to be in the three "best" models, however the model for set #18, though not the best among the other models found in the same search, still outperformed the models for set #18 found through the prior DSPSA searches, which either fixed the inputs and/or fixed the n_k values. Directly comparing the models found by exhaustive search and DSPSA with feature selection and n_k search for set #18 (Table 4.3), allowing for both expanded searches increased the best model's NRMSE fits by about 20% or more for each part of the objective function (estimation, validation, and overall data).

Method	Avg. Est. NRMSE %	Avg. Val. NRMSE %	Avg. Overall NRMSE %
Exhaustive Search	50.97	56.63	46.03
DSPSA w/ feature selection and n_k	83.02	76.09	76.85

Table 4.3: *Just Walk*: Participant 230 "Full Search" ARX Model Comparison for Est/Val Set # 18

Step responses for the three best models (sets 11, 13, and 16) demonstrating the behavior response (steps) for Participant 230 given a step increase in goals are shown in Figure 4.13. All three models show similar step responses, with little variation in terms of time constant and gain. Each model has a gain of about 1, meaning that the participant is expected to increase their step count by about 1 for every 1 step increase in their daily goals.

Additional plots to further demonstrate the results of DSPSA, additional plots for the model with the highest fits can be found in Appendix C. This model is demonstrated on the overall data and the estimation and validation data in Figures C.1 and C.2, respectively. The fits of each model tested at each DSPSA iteration for set

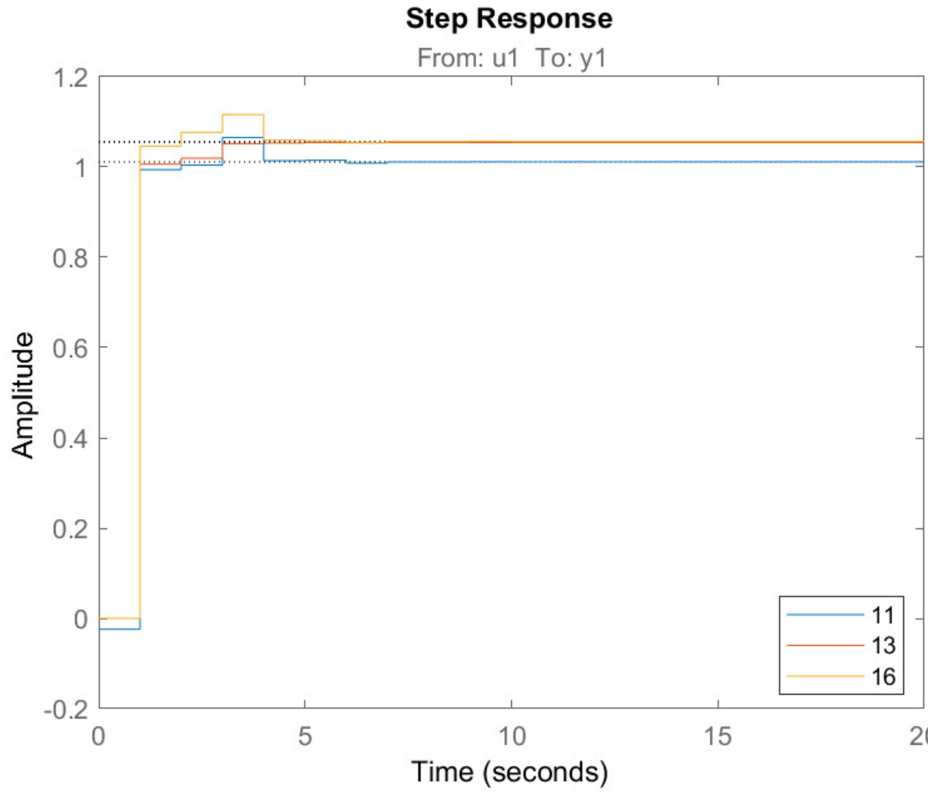

Figure 4.13: *Just Walk*: Participant 230 Behavior Response (steps) to a Step Increase in Goals

#18 are also shown in Figure C.3. The feature selection is shown in Figure C.4. Unlike the prior cases, the features quickly converged, as most models after the first few iterations used the same three inputs: *Goals, Expected Points* and *Granted Points.* The iterations for the regressor structure search are demonstrated in Figures C.5 - C.7.

4.5 *Just Walk* Participant 008: Feature Selection with n_a, n_b, and n_k Search in ARX Models with DSPSA

Although the full search (n_a, n_b, and n_k with feature selection) produces fits of about 70-80% for Participant 230 (the operant learner), such is not the case with most participants. Other participants have shown much greater variability in their behavioral responses, which is reflected in the low fit of models created using their data. Despite the lower fits, using DSPSA to optimize models of walking behavior is still a structured technique to quickly find the parameters of best or near-best models, given an individual's data. In this section, a full search is illustrated for Participant 008.

			NRMSE Fit (%)					Avg Est	Avg Val	Avg Overall	Obj Fxn
	E*	V*	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	%	%	%	%
1	[1,2]	[3,4,5]	15.64	47.68	55.04	38.75	23.58	31.66	39.12	23.22	25.37
2	[1,3]	[2,4,5]	29.20	35.89	54.63	42.72	37.90	41.91	38.84	31.91	34.75
3	[1,4]	[2,3,5]	33.44	50.15	49.35	72.17	41.47	52.80	46.99	33.31	36.60
4	[1,5]	[2,3,4]	14.23	41.35	57.48	44.05	48.46	31.34	47.63	36.69	39.56
5	[2,3]	[1,4,5]	20.16	55.33	49.83	36.09	33.19	52.58	29.81	30.27	31.46
6	[2,4]	[1,3,5]	28.99	44.19	51.76	55.22	37.92	49.71	39.55	33.46	35.91
7	[2,5]	[1,3,4]	7.97	37.86	56.77	54.57	45.94	41.90	39.77	35.23	39.08
8	[3,4]	[1,2,5]	-20.04	39.00	63.21	66.35	41.29	64.78	20.08	34.12	29.54
9	[3,5]	[1,2,4]	-22.74	37.80	59.27	35.04	45.19	52.23	16.70	33.93	22.94
10	[4,5]	[1,2,3]	-13.14	33.69	55.45	50.79	48.65	49.72	25.33	34.15	25.47
11	[3,4,5]	[1,2]	-29.07	51.06	61.08	35.41	17.42	37.97	10.99	17.15	18.75
12	[2,4,5]	[1,3]	23.80	37.56	57.95	58.44	45.60	47.20	40.87	32.90	38.79
13	[2,3,5]	[1,4]	18.41	26.37	54.15	55.74	40.41	40.31	37.08	33.71	39.34
14	[2,3,4]	[1,5]	22.16	60.99	60.67	59.61	34.47	60.42	28.31	35.23	31.94
15	[1,4,5]	[2,3]	15.20	45.61	58.84	55.52	44.19	38.31	52.22	37.42	40.32
16	[1,3,5]	[2,4]	19.29	48.95	57.37	43.63	47.53	41.40	46.29	37.43	40.47
17	[1,3,4]	[2,5]	26.37	39.30	58.67	50.32	38.87	45.12	39.09	35.71	35.97
18	[1,2,5]	[3,4]	-13.69	51.30	59.96	61.94	41.38	26.33	60.95	35.86	49.87
19	[1,2,4]	[3,5]	35.00	61.78	57.25	59.64	33.01	52.14	45.13	37.49	37.30
20	[1,2,3]	[4,5]	26.63	46.86	67.21	38.16	33.56	46.90	35.86	34.03	32.02

Figure 4.14: *Just Walk*: Participant 008 Results for ARX Models found by Discrete SPSA with Feature Selection and n_a, n_b and n_k Search (Best cases highlighted in yellow)

As for Participant 230, all eight original inputs were initialized for feature selection. All three regressor structure orders were also initialized for all outputs (n_a: 1) and all inputs (n_b, n_k: 8). The same bounds were used, and so n_a and n_b were constrained between 1 and 3, while n_k was bound between 0 and 1. For each set of estimation and validation data, 110 iterations of DSPSA were implemented. The total search took 17.9 minutes.

		FEATURES								ARX		
	Total Inp	Goals	Exp Pts	Granted Pts	PredBusy	PredStress	PredTyp	Weekend	Temp	na	nb	nk
1	4	x		x				x	x	1	3323	1001
2	3	x	x	x						2	222	000
3	5	x	x	x	x	x				1	22231	00010
4	4	x	x	x		x				1	3111	1101
5	4	x	x	x		x				1	2222	0001
6	4	x	x	x		x				2	3111	0100
7	3	x	x	x						3	311	110
8	3	x	x	x						2	233	000
9	4	x	x	x		x				2	2331	1000
10	3	x	x	x						1	322	100
11	3	x	x	x						2	332	101
12	6	x	x	x	x			x	x	3	222121	010001
13	5	x	x	x	x			x		1	22121	00000
14	5	x	x	x	x			x		1	33331	00001
15	5	x	x	x		x		x		2	33311	00010
16	5	x	x	x		x		x		1	23311	11011
17	4	x	x	x		x				1	3311	0001
18	5	x	x	x		x		x		3	31311	00001
19	5	x	x	x		x		x		2	32332	00000
20	5	x	x	x		x		x		3	31313	01001

Figure 4.15: *Just Walk*: Participant 008 Feature Selection for Each Set of Estimation and Validation Data with Corresponding n_a, n_b and n_k Values

From the table in Figure 4.14, top three models (as evaluated by the objective function in Equation 4.1) were the models obtained in sets 15, 16, and 18, as highlighted in yellow. The model obtained in set 18 demonstrated the highest weighted fit at 49.9%. All three models used five features, as outlined in Figure 4.15. They each used *Goals, Expected Points, Granted Points, Predicted Stress* and *Weekend.* The corresponding n_a, n_b and n_k values are also shown in the same figure.

The step response for Participant 008's behavior (steps) to a step increase in goals is also shown in Figure 4.16 for the top three models. Compared to Participant 230, there is more variability in the magnitude of the step responses across the three models. However, all three step responses are positively valued, meaning that Participant 008 is expected to increase their step count given an increase in their goals, although

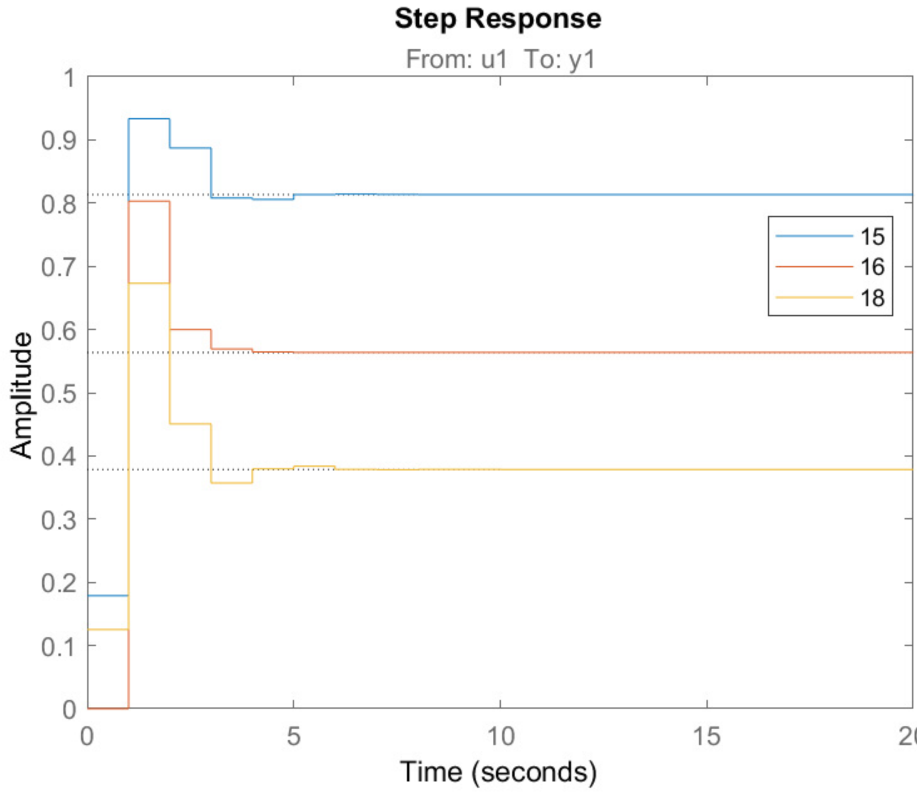

Figure 4.16: *Just Walk*: Participant 008 Behavior Response (steps) to a Step Increase in Goals

the increase will not be one to one. Additional plots to illustrate the use of DSPSA to find models of walking behavior for Participant 008 are available in Appendix D.

4.6 *Just Walk* Participant 057: Feature Selection with n_a, n_b, and n_k Search in ARX Models with DSPSA

To demonstrate this method on a third participant, the data from Participant 057 was also used to show the use of DSPSA to optimize personalized ARX models of their individual behavior. The models were obtained using the same full search procedure that was performed for Participant 230 and Participant 008. This includes the same initialized features and regressor orders, as well as the same bounds. For Participant 057, 110 iterations per set of estimation and validation data were used, which took a total of 16.8 minutes to compute.

			NRMSE Fit (%)					Avg Est	Avg Val	Avg Overall	Obj Fxn
	E*	V*	Cycle 1	Cycle 2	Cycle 3	Cycle 4	Cycle 5	%	%	%	%
1	[1,2]	[3,4,5]	36.97	52.75	47.89	16.99	15.20	44.86	26.69	26.29	23.19
2	[1,3]	[2,4,5]	25.31	3.95	48.18	27.35	21.49	36.75	17.60	19.80	18.49
3	[1,4]	[2,3,5]	35.96	10.47	28.98	31.16	35.50	33.56	24.98	16.74	19.43
4	[1,5]	[2,3,4]	33.88	10.51	16.85	24.99	43.00	38.44	17.45	26.53	20.54
5	[2,3]	[1,4,5]	30.45	31.35	26.51	24.49	33.29	28.93	29.41	19.69	19.55
6	[2,4]	[1,3,5]	25.15	49.66	34.51	36.66	25.16	43.16	28.27	24.80	21.82
7	[2,5]	[1,3,4]	25.20	28.80	22.85	31.44	37.33	33.07	26.50	25.72	22.34
8	[3,4]	[1,2,5]	28.25	23.51	38.22	38.38	30.03	38.30	27.26	25.32	23.49
9	[3,5]	[1,2,4]	29.62	23.96	21.02	31.82	36.74	28.88	28.47	26.13	25.63
10	[4,5]	[1,2,3]	18.37	6.58	9.96	27.49	38.34	32.92	11.64	20.72	14.87
11	[3,4,5]	[1,2]	29.05	20.96	23.97	34.86	38.57	32.47	25.01	27.31	25.70
12	[2,4,5]	[1,3]	29.68	45.37	37.46	48.18	34.99	42.85	33.57	32.11	24.60
13	[2,3,5]	[1,4]	32.34	32.92	27.32	27.16	32.73	30.99	29.75	28.46	23.70
14	[2,3,4]	[1,5]	38.76	42.12	37.88	30.57	23.55	36.86	31.15	26.85	25.43
15	[1,4,5]	[2,3]	37.87	17.62	30.79	37.16	37.86	37.63	24.20	26.38	23.44
16	[1,3,5]	[2,4]	33.65	16.55	25.86	24.17	34.38	31.30	20.36	25.56	21.70
17	[1,3,4]	[2,5]	41.90	20.26	44.32	21.66	24.95	35.96	22.60	21.43	20.18
18	[1,2,5]	[3,4]	40.06	32.30	23.45	28.10	38.33	36.90	25.77	26.75	24.24
19	[1,2,4]	[3,5]	31.59	40.98	30.03	34.93	25.96	35.83	27.99	27.49	25.70
20	[1,2,3]	[4,5]	24.98	26.77	32.28	13.49	23.76	28.01	18.63	21.44	19.95

Figure 4.17: *Just Walk*: Participant 057 Results for ARX Models found by Discrete SPSA with Feature Selection and n_a, n_b and n_k Search (Best cases highlighted in yellow)

The results of this search are shown in Figure 4.17, with the top three models, corresponding to data sets 9, 11, and 19, highlighted in yellow. The inputs and regressor orders for each model are also outlined in Figure 4.18. These models were evaluated based on the weighted fit as described in Equation 4.1. The best models obtained by the algorithm are the ones for sets 11 and 19, which have the same weighted fit (25.7%).

The models fits obtained for Participant 057 are generally lower than the fits obtained for both Participant 230 and Participant 008. However, this is due to limitations within the data provided by the participant during the *Just Walk* intervention and not a result of poor parameter value assignment or choice of features.

		FEATURES								ARX		
	Total Inp	Goals	Exp Pts	Granted Pts	PredBusy	PredStress	PredTyp	Weekend	Temp	na	nb	nk
1	6	x	x	x	x		x	x		3	321111	000100
2	5		x	x	x			x	x	2	22131	00100
3	5		x	x	x			x	x	3	12232	10100
4	7	x	x	x	x		x	x	x	2	2112212	0001100
5	6		x	x	x		x	x	x	1	123112	001100
6	5	x		x	x	x		x		2	22131	10110
7	5	x		x	x	x		x		1	21321	10000
8	4	x		x	x			x		3	1221	1010
9	4	x		x	x			x		1	1121	1010
10	4	x		x	x			x		1	2112	0000
11	4	x		x	x			x		2	2131	1000
12	5	x	x	x	x			x		1	23133	10011
13	4	x		x	x			x		1	2321	1010
14	5	x		x	x			x	x	2	23321	10101
15	5	x		x	x			x	x	1	31212	10101
16	5	x		x	x			x	x	1	21311	00001
17	3	x		x	x			x	x	3	11311	10010
18	5	x		x	x			x	x	3	22211	10101
19	5	x		x	x			x	x	2	23132	10011
20	5	x		x	x			x	x	1	12223	00100

Figure 4.18: *Just Walk*: Participant 057 Feature Selection for Each Set of Estimation and Validation Data with Corresponding n_a, n_b and n_k Values

Figure 4.19 illustrates the step responses for the best three models which were the ones obtained for sets 9, 11, and 19. Like the previous two participants all three step responses are positive valued, suggesting that Participant 057 increases their steps in response to an increase in step goals. However, the amplitude of these step responses are the lowest of the three participants, which suggests that Participant 057 is the least responsive of the three. Additional plots to further illustrate the DSPSA results for Participant 057 are in Appendix E.

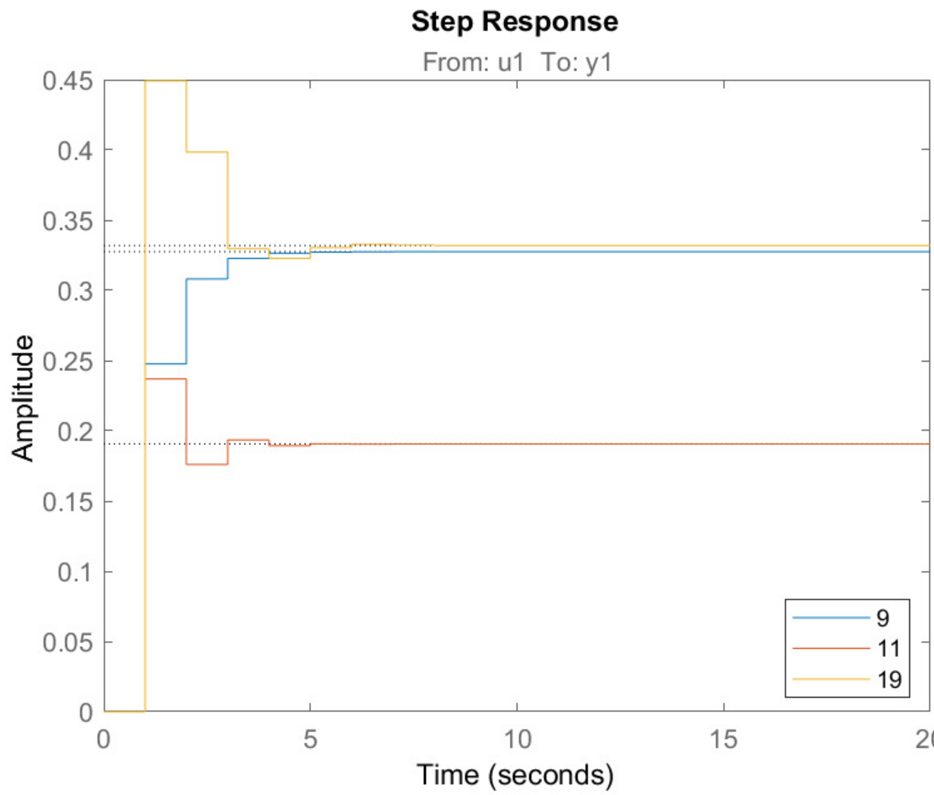

Figure 4.19: *Just Walk*: Participant 057 Behavior Response (steps) to a Step Increase in Goals

4.7 Summary and Conclusions

This chapter has demonstrated the use of DSPSA to simultaneously search over regressor structures and features, to find the near-optimal ARX models of individual walking behavior given intervention data. The method presented not only illustrates the influence of data selection on model structure, but further highlights that DSPSA is much more efficient than exhaustive search.

First, by using DSPSA to search over n_a and n_b orders for a fixed 4-input model with $n_k = 1$ for all inputs, DSPSA found models comparable to the models for the same participant (230) as found by exhaustive search from a previous published study Freigoun *et al.* (2017). Not only did DSPSA find models with comparable fits, but it was also able to find these models within 10 minutes with 70 iterations per data set, being run on a local computer, which is much more computationally efficient than the exhaustive search, which can take hours to days, depending on the size of the search space. For the search parameters used in Freigoun *et al.* (2017) (n_a and n_b search only), the exhaustive search was estimated to take about 25 to 30 minutes. However, this time increases drastically as the search space increases in size. Adding search over $n_k \in \{0, 1\}$ increases the number of regressor order combinations to 3,888, which is 16 times the number of combinations with a search over only $n_a,\ n_b \in \{1, 3\}$. With searching over 3,888 regressor order combinations for 20 possible sets of data, this increases the total number of models to evaluate by 72,900. Running an exhaustive search over n_a, n_b, and n_k with four fixed inputs on the same laptop used to perform the DSPSA searches illustrated in this chapter took 8.7 hours. To also search over features, this process would then have to be repeated for every combination of fixed inputs. Using the same participant data, the DSPSA application addressed this expanded search by allowing for both feature selection and search over n_k. However, allowing n_k to be 0 and for search over features, DSPSA was able to find models for every set of estimation and validation data that outperformed the original search. This search was also faster than exhaustive search, requiring only 13.6 minutes, highlighting both the efficiency and flexibility of DSPSA, especially as the search space increases.

Additional demonstrations were shown for other participants of *Just Walk* including Participant 008 and Participant 057, which which although obtained lower fits than for Participant 230, further highlight the efficiency of using DSPSA to optimize model parameters and features. The difference in the fits obtained by models across participants is likely due to limitations of the data itself. Further research should be done to translate these models into control systems to then optimize the interventions themselves.

Although MoD was explored in previous chapters as a modeling alternative to ARX, using DSPSA to optimize ARX models in this chapter demonstrates a valuable "near-optimal" estimation framework to assess model parameters and features faster than exhaustive search. This framework is necessary for future studies that implement these models to determine interventions (i.e. when a participant would be most responsive, what goals they would be responsive to, etc.), such as in the *YourMove* study, which is currently underway at UCSD and ASU and has about 400 participants. This would require estimating models for large numbers of participants within a short time span, which can be addressed with DSPSA.

Chapter 5

CONCLUSIONS AND FUTURE WORK

This thesis has presented the development of idiographic models of walking behavior for personalized mHealth interventions using Model-on-Demand (MoD) and ARX estimation with Discrete Simultaneous Perturbation Stochastic Approximation (DSPSA) as a selection technique. MoD estimation outperformed classical linear ARX models on both simulated systems with noise (CSTR) and real data obtained from participants of *Just Walk* and *HeartSteps*. As an adaptive modeling technique, MoD is better able to address the nonlinearity of walking behavior as well as the presence of noise, which may arise from a number of sources including participant self-reported measures and disturbances or unmeasured influences. This is especially significant as idiographic modeling demonstrated in this thesis is a "small data" problem, which has distinct obstacles to generalizing models, even for a given individual's behavior. Despite the availability of data-gathering technologies, there still remains limitations to the amount of data available and restrictions to our ability to obtain data for a single individual.

Although Model-on-Demand estimation obtains better models of individual walking behavior than ARX (evaluated by the normalized root mean squared error), MoD is more difficult to implement, as there are more parameters that must be specified, which requires either *a priori* knowledge or exhaustive search. So, to overcome the "barriers to use" of MoD, DSPSA was used to efficiently and simultaneously search over model parameters and features to obtain near-optimal models. DSPSA is simple to implement and presents a structured method to optimize MoD models. By simultaneously searching over model features and MoD parameters, MoD models that outperformed ARX were obtained within a few minutes and without *a priori* knowledge necessary from the MoD user (i.e. researchers or intervention developers). As such, coupling MoD with DSPSA makes the technique scalable in the sense that (1) it can be used by many individuals without significant training and (2) it can find models quickly, reducing both user and computational effort, so that MoD models can be developed for many individuals simultaneously.

Given the small data approach taken in this research, another issue becomes how we evaluate the models, especially since there is limited data available modeling. As discussed in Chapters 2 and 3, the behavior data was initially split into the first 75% as estimation data and the last 25% validation data. However, this may be insufficient to evaluate the accuracy of the model, especially given the evolving and time-dependent nature of behavior. So, to address this, we explored variations of partitioning an individual's data into estimation and validation data and the corresponding development of models from that data. While the focus of this study was ARX modeling, using discrete SPSA to search over ARX regressor orders and feature selection for models using the *Just Walk* data demonstrated that traditional segmentation of data into estimation and validation regions may not be optimal. Multiple models with high fit percentages were found using DSPSA on the *Just Walk* data from various partitions of estimation and validation data. Since *Just Walk* is an intervention designed using system identification principles in the form of a repeated multisinusoidal signal (a 16 day signal, repeated for 5 cycles, for a total of 80 days), this demonstrates how an intervention designed using system identification can be used to produce quality data, which can then be used to create more accurate models of participant behavior. The modeling results shown in this thesis using the data produced from *Just Walk* also shows much promise for the development of more accurate idiographic models from data obtained using system identification principles. These models could then be used to design more effective control systems for personalized interventions.

The application of MoD and DSPSA to data from both *HeartSteps* and *Just Walk* demonstrates that models of individual walking behavior can be efficiently produced to then be used for designing control systems to optimize the impact of the intervention for its participants. This is done by personalization, as control systems would then be developed using models of behavior created from the individual's own behavior, thereby tailoring the intervention to their particular habits and idiosyncrasies to maximize receptivity and responsiveness. Further studies should be performed using models developed through MoD and DSPSA to confirm whether or not MoD models can provide better interventions than simpler, ARX models. As mentioned previously, using DSPSA also makes the process of developing MoD models for individuals scalable. However, further research should also be done to improve the objective function used in DSPSA and consider what objectives should be met to ensure effective intervention design and what practical constraints should be accounted for.

But despite the continued research that should be done to further explore the use of DSPSA as well as the implementation of DSPSA-optimized models in intervention design, the use of DSPSA presents a novel approach to model development for personalized interventions. Other considerations such as modeling technique (i.e. the use of Model-on-Demand or AutoRegressive with eXogenous input) and partitioning the individual's data into estimation and validation data sets, also highlight the benefits of implementing DSPSA, as the simulation-based stochastic search mechanism is able to efficiently and simultaneously search over model features and parameters to find optimal models, addressing many design aspects that researchers must consider when deriving models from data. DSPSA may also become more necessary as search spaces expand, since there may be many other measured factors to include or model parameters to be defined as future studies collect more environmental and physiological data or explore other modeling techniques. However, with DSPSA as an efficient search algorithm to quickly define and evaluate idiographic models, research should also be done to translate these models into control systems to optimize the impacts of the intervention.

REFERENCES

Aksakalli, V. and M. Malekipirbazari, "Feature selection via binary simultaneous perturbation stochastic approximation", Pattern Recognition Letters **75**, 41–47 (2016).

Bailey, D. and D. E. Rivera, "Input design gui", (2020).

Bequette, B. W., in "Process Dynamics: Analysis and Simulation", (Prentice-Hall, 1998), URL `https://www.rpi.edu/dept/chem-eng/WWW/faculty/bequette/education/links_mods/cstr/`.

Braun, M. W., *Model-on-demand nonlinear estimation and model predictive control: Novel methodologies for process control and supply chain management* (Arizona State University, 2001).
Modeling,

Braun, M. W., D. E. Rivera and A. Stenman, "A Model-on-Demand identification methodology for nonlinear process systems", Int. J. Control **74**, 18, 1708 – 1717 (2001).

Freigoun, M. T., C. A. Martín, A. B. Magann, D. E. Rivera, S. S. Phatak, E. V. Korinek and E. B. Hekler, "System identification of just walk: A behavioral mhealth intervention for promoting physical activity", in "2017 American Control Conference (ACC)", pp. 116–121 (IEEE, 2017).

Hekler, E. B., D. E. Rivera, C. A. Martin, S. S. Phatak, M. T. Freigoun, E. Korinek, P. Klasnja, M. A. Adams and M. P. Buman, "Tutorial for using control systems engineering to optimize adaptive mobile health interventions", Journal of medical Internet research **20**, 6, e8622 (2018).

Kha, R. T., D. E. Rivera, P. Klasnja and E. Hekler, "Model personalization in behavioral interventions using model-on-demand estimation and discrete simultaneous perturbation stochastic approximation", in "2022 American Control Conference", pp. 671–676 (IEEE, 2022).

Klasnja, P., S. Smith, N. J. Seewald, A. Lee, K. Hall, B. Luers, E. B. Hekler and S. A. Murphy, "Efficacy of contextually tailored suggestions for physical activity: A micro-randomized optimization trial of heartsteps", Annals of Behavioral Medicine **53**, 6, 573–582 (2019).

Martin, C. A., D. E. Rivera, E. B. Hekler, W. T. Riley, M. P. Buman, M. A. Adams and A. B. Magann, "Development of a control-oriented model of social cognitive theory for optimized mhealth behavioral interventions", IEEE Transactions on Control Systems Technology **28**, 2, 331–346 (2018).

Nandola, N. N. and D. E. Rivera, "Model-on-Demand predictive control for nonlinear hybrid systems with application to adaptive behavioral interventions", in "49th IEEE Conference on Decision and Control (CDC)", pp. 6113–6118 (IEEE, 2010).

Phatak, S. S., M. T. Freigoun, C. A. Martín, D. E. Rivera, E. V. Korinek, M. A. Adams, M. P. Buman, P. Klasnja and E. B. Hekler, "Modeling individual differences: A case study of the application of system identification for personalizing a physical activity intervention", Journal of biomedical informatics **79**, 82–97 (2018).

Rivera, D. E., E. Hekler, J. S. Savage and D. Symons Downs, "Intensively adaptive interventions using control systems engineering: Two illustrative examples", in "Optimization of Behavioral, Biobehavioral, and Biomedical Interventions", edited by L. M. Collins and K. C. Kugler, pp. 121–173 (Springer, 2018).

Schwartz, J. D., W. Wang and D. E. Rivera, "Simulation-based optimization of process control policies for inventory management in supply chains", Automatica **42**, 8, 1311–1320, URL `https://www.sciencedirect.com/science/article/pii/S0005109806001464` (2006).

Spall, J. C., "An overview of the simultaneous perturbation method for efficient optimization", Johns Hopkins APL Technical Digest **19**, 4, 482–492 (1998).

Spruijt-Metz, D., B. M. Marlin, M. Pavel, D. E. Rivera, E. Hekler, S. De La Torre, M. El Mistiri, N. M. Golaszweski, C. Li, R. Braga De Braganca *et al.*, "Advancing behavioral intervention and theory development for mobile health: The heartsteps ii protocol", International journal of environmental research and public health **19**, 4, 2267 (2022).

Stenman, A., *Model on demand: Algorithms, analysis and applications* (Citeseer, 1999).

Wang, Q. and J. C. Spall, "Discrete simultaneous perturbation stochastic approximation for resource allocation in public health", in "2014 American Control Conference", pp. 3639–3644 (IEEE, 2014).

APPENDIX A

CSTR SIMULINK MODEL

This CSTR simulation was set up as per Bequette (1998). In this model, there two measured outputs were considered (1) C_A, the concentration of reactant A in the tank, and (2) T, the temperature of the contents within the tank. The concentration of reactant A, C_A, evolves as:

$$\frac{dC_A}{dt} = \frac{F}{V}(C_{Af} - C_A) - r \tag{A.1}$$

in which F is the volumetric flow of both the feed and exit streams, V is the volume of the tank, C_{Af} is the concentration of A in the feed stream, and r is the rate of reaction, defined as:

$$r = k_o \exp \frac{-\Delta E}{RT} C_A \tag{A.2}$$

where T is the temperature inside the tank. The differential equation describing tank temperature is:

$$\frac{dT}{dt} = \frac{F}{V}(T_f - T) - \frac{\Delta H}{C_p \rho} r - \frac{UA}{C_p \rho V}(T - T_j) \tag{A.3}$$

in which T_f is the temperature of the feed stream, ΔH is the change in enthalpy, C_p is the heat capacity, ρ is the density, and T_j is the jacket temperature.

These differential equations were implemented in Simulink as per Figure A.1.

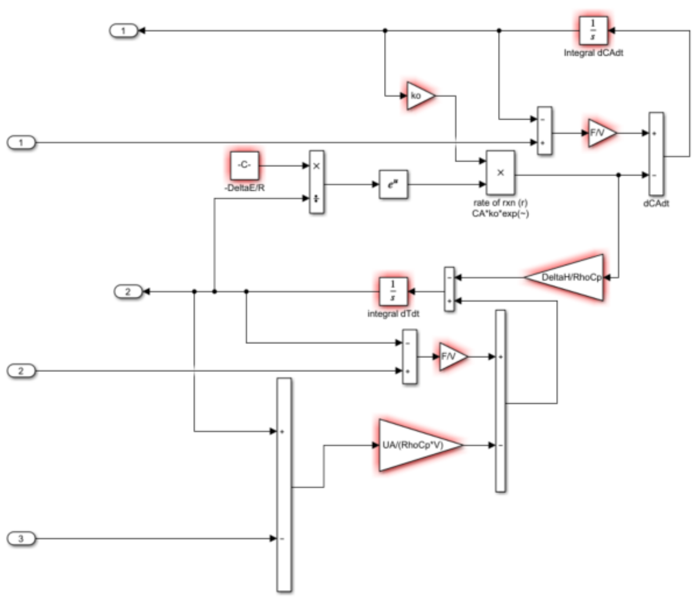

Figure A.1: CSTR Differential Equations Modeled in Simulink

To introduce noise into the system, random number generator blocks were implemented with zero mean and a variance of one. Gain blocks were also added and used to vary the amount of noise relative to the inputs and outputs of interest. Both process and measurement noise were added to the feed temperature and feed concentration. Only measurement noise was added to the jacket temperature, output concentration, and tank temperature.

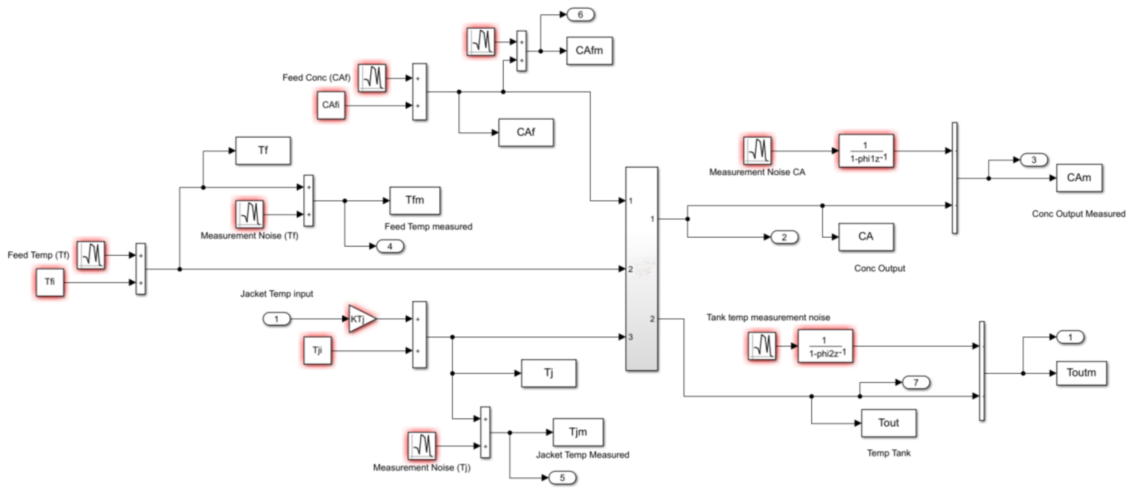

Figure A.2: CSTR Measurement and Process Noise Modeled in Simulink

The reactor parameters are provided in Table A.1, and the steady state values are noted in Table A.2.

Parameter	Value/Type
F	$1\ m^3/hr$
V	$1\ m^3$
k_o	$14825{\times}3600\ 1/hr$
ΔH	$-5215\ kcal/kgmol$
ΔE	$11843\ kcal/kgmol$
ρC_p	$500\ kcal/(m^3\ C)$
UA	$250\ kcal/(m^3\ C\ hr)$
R	$1.98589\ kcal/(kgmol\ K)$

Table A.1: CSTR Reactor Parameters

Parameter	Value/Type
$C_{A,s}$	$7.5709\ m^3/hr$
T_s	$315.0402\ K$

Table A.2: CSTR Steady State Values

APPENDIX B

CSTR JACKET TEMPERATURE SIGNAL DESIGN

The parameters used to create the multisine input signal for the jacket temperature in the CSTR simulation are documented in Table B.1. Two realizations of the signal with these same parameters were used to obtain distinct sets of estimation and validation data. These signals were created using the Input Design GUI created by Daniel Bailey and Daniel E. Rivera. One realization of the signal is shown in Figure B.1.

Parameter	Value/Type
Type	Multi Sinusoidal
Sampling Time	0.5
Amplitude (±)	1
Cycles	2
α	2
β	3
τ_{low}	2.5 hr
τ_{high}	3.5 hr
Generation Method	Minimum Crest Factor

Table B.1: CSTR Signal Design Parameters

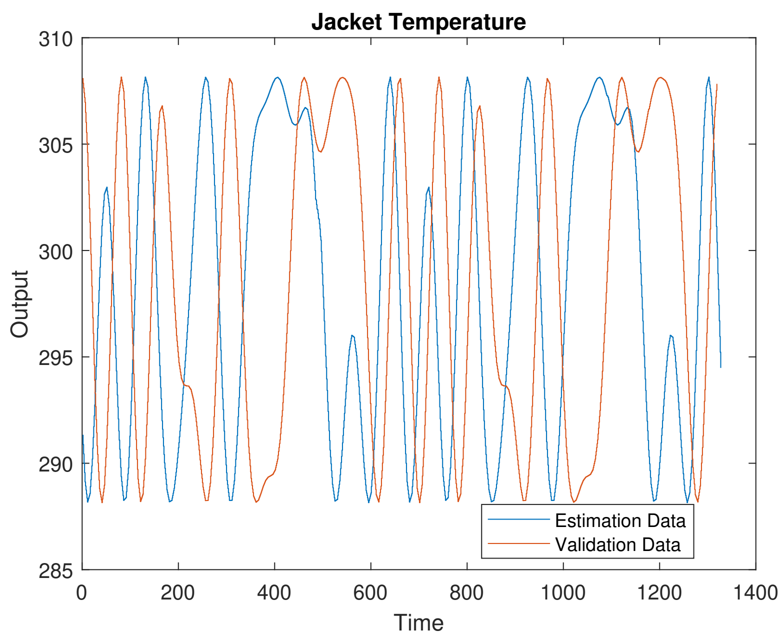

Figure B.1: Signals Produced in the Input Design GUI for CSTR Simulation (reprinted)

The signal in Figure B.1 was multiplied by a user-specified gain to provide sufficient variation in jacket temperature. Additional plots to further document properties of this signal are shown in Figures B.2 - B.4, which include the power spectral density, the autocorrelation, and a histogram of the frequency of the signal amplitudes.

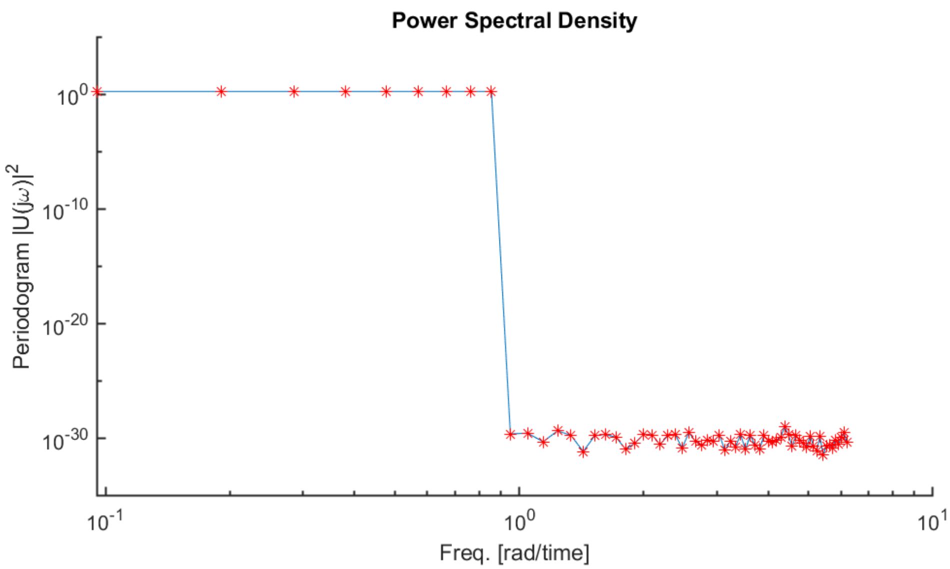

Figure B.2: Power Spectral Density of the Input Signal for CSTR Simulation

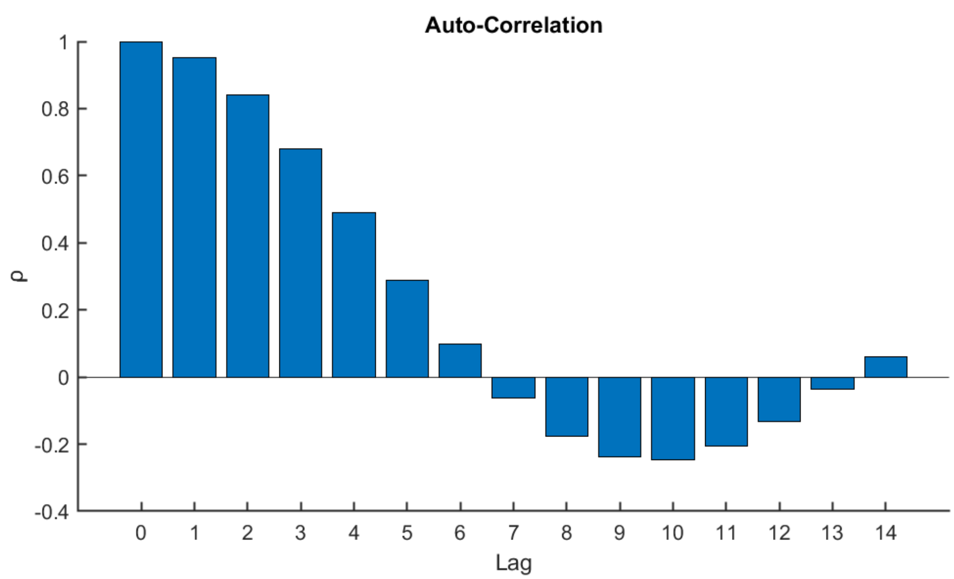

Figure B.3: Autocorrelation of the Input Signal for CSTR Simulation

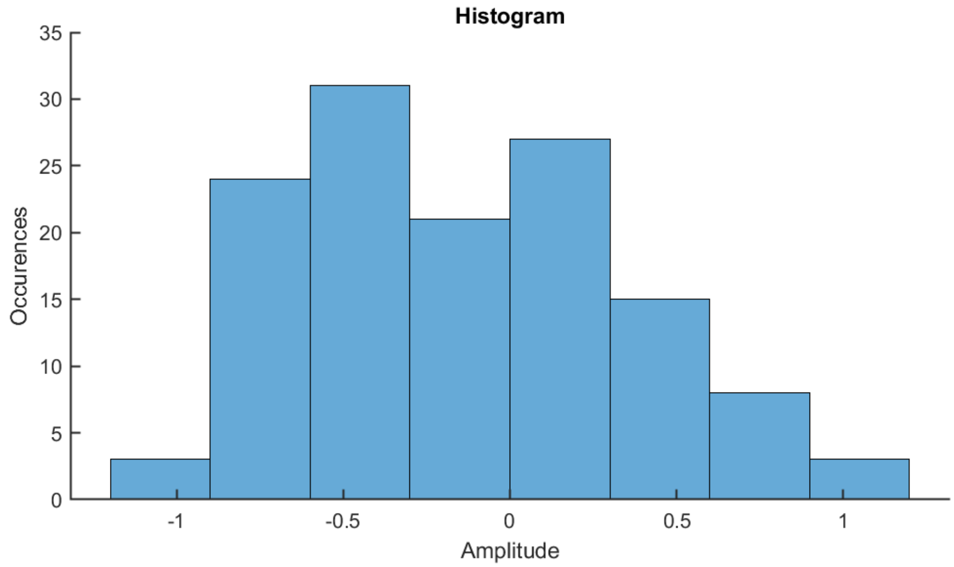

Figure B.4: Histogram of the Input Signal for CSTR Simulation

All plots were provided by the Input Design GUI (Bailey and Rivera, 2020).

APPENDIX C

ARX ESTIMATION WITH DSPSA: PARTICIPANT 230 FULL SEARCH ADDITIONAL PLOTS

The following plots are the DSPSA results for set #13 from Figure 4.11 and 4.12, which had the highest weighted fit of 82.16%. To reiterate, the inputs of this model are outlined in Table C.1.

Initialized Features	Selected Features	n_b	n_k
Goals	Goals	2	1
Expected Points	Expected Points	2	1
Granted Points	Granted Points	1	0
Predicted Busyness	-	-	-
Predicted Stress	-	-	-
Predicted Typical	-	-	-
Weekend	-	-	-
Temperature	-	-	-

Table C.1: Feature Selection and n_b, n_k orders ($n_a = 2$)

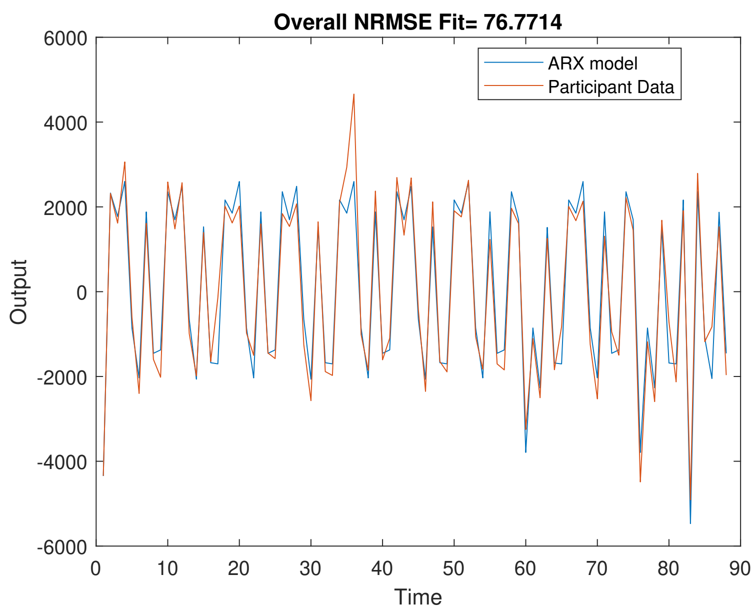

Figure C.1: Participant 230: ARX Models with Feature Select on Overall Data for Set #13

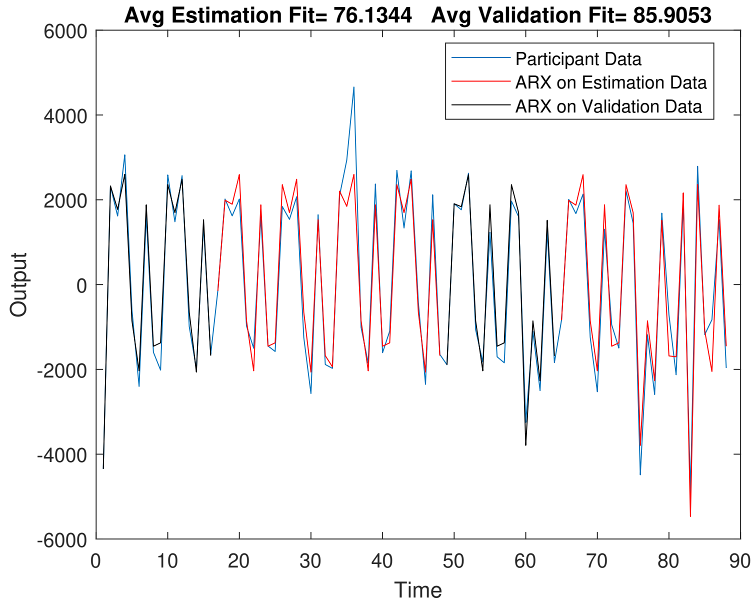

Figure C.2: Participant 230: ARX Models with Feature Select on Estimation and Validation Data for Set #13

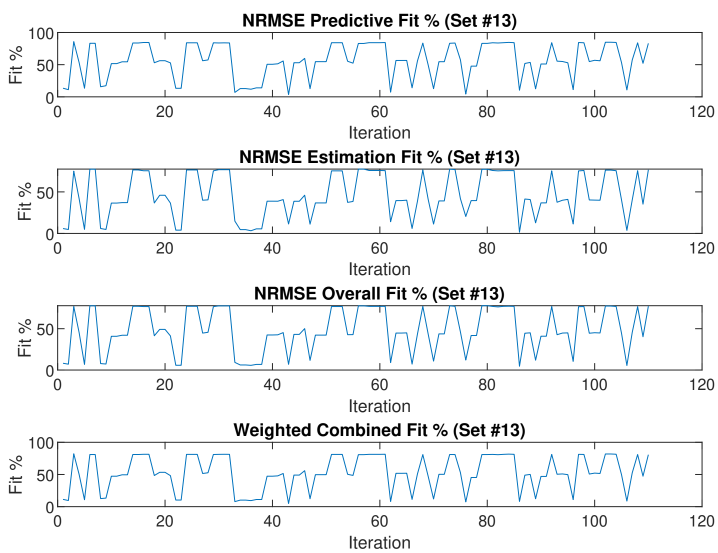

Figure C.3: Participant 230: Fit (%) Iterations for ARX Models with Feature Select and n_k search, Set #13

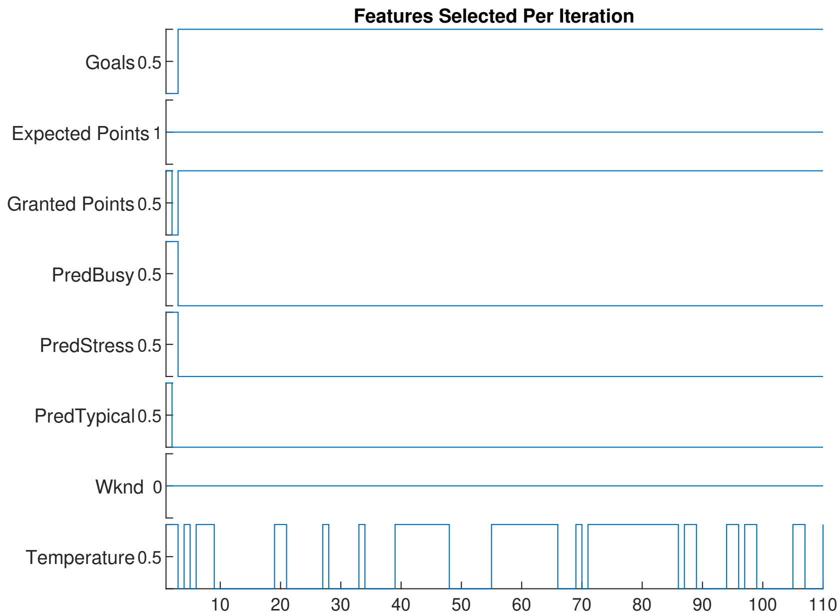

Figure C.4: Participant 230: Feature Select Iterations for ARX Models with Feature Select and n_k search, Set #13

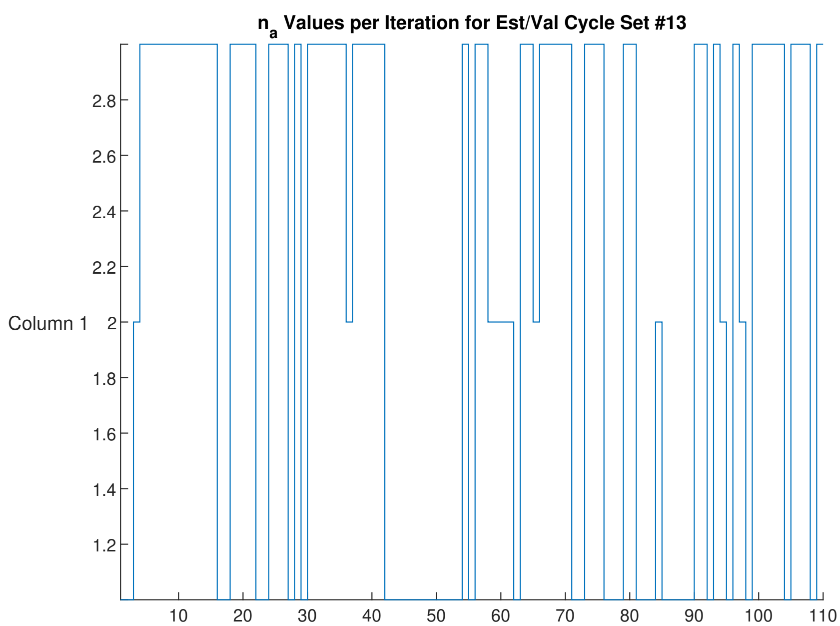

Figure C.5: Participant 230: n_a Iterations for ARX Models with Feature Select and n_k search, Set #13

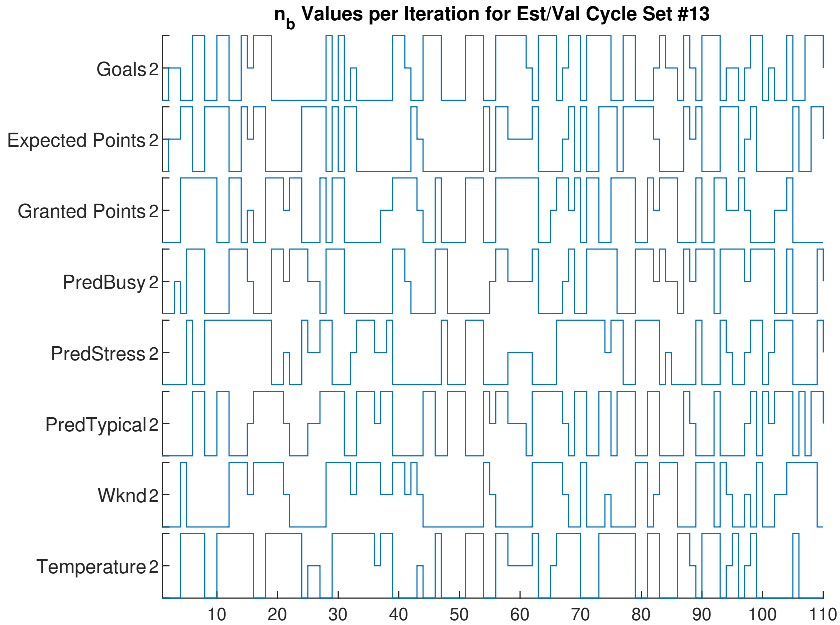

Figure C.6: Participant 230: n_b Iterations for ARX Models with Feature Select and n_k search, Set #13

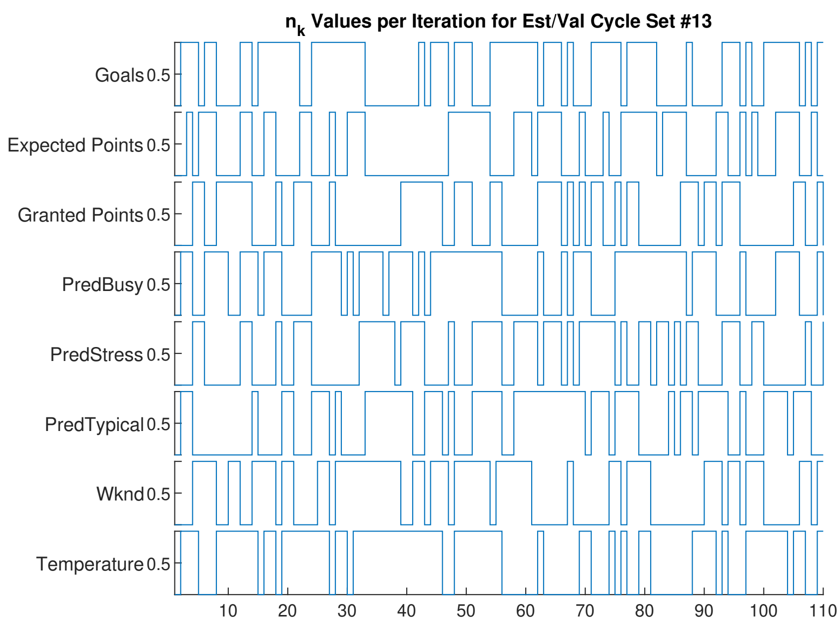

Figure C.7: Participant 230: n_k Iterations for ARX Models with Feature Select and n_k search, Set #13

APPENDIX D

ARX ESTIMATION WITH DSPSA: PARTICIPANT 008 FULL SEARCH ADDITIONAL PLOTS

The following plots are the DSPSA results for set #18 from Figure 4.14 and 4.15, which had the highest weighted fit. To reiterate, the inputs of this model are outlined in Table D.1.

Initialized Features	Selected Features	n_b	n_k
Goals	Goals	3	0
Expected Points	Expected Points	2	0
Granted Points	Granted Points	3	0
Predicted Busyness	-	-	-
Predicted Stress	Predicted Stress	3	0
Predicted Typical	-	-	-
Weekend	Weekend	2	0
Temperature	-	-	-

Table D.1: Participant 008: Feature Selection and n_b, n_k orders ($n_a = 3$)

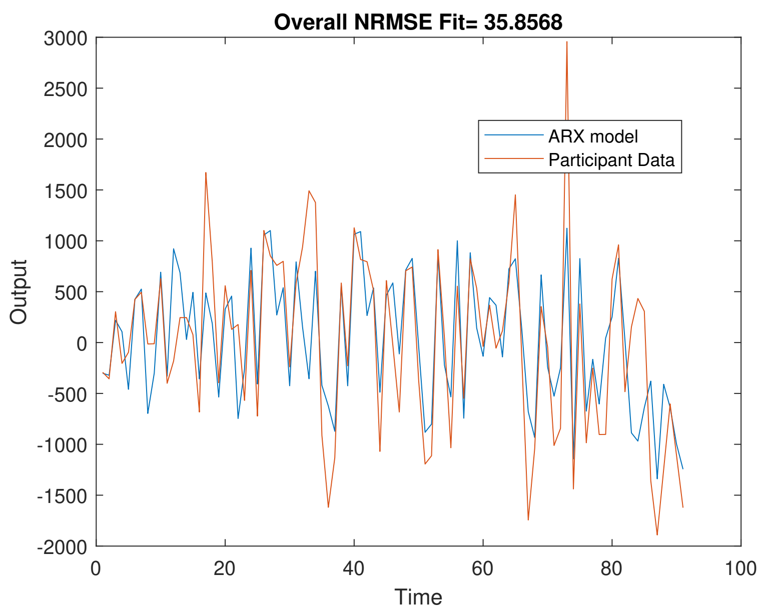

Figure D.1: Participant 008: ARX Model with Feature Select on Overall Data for Set #18

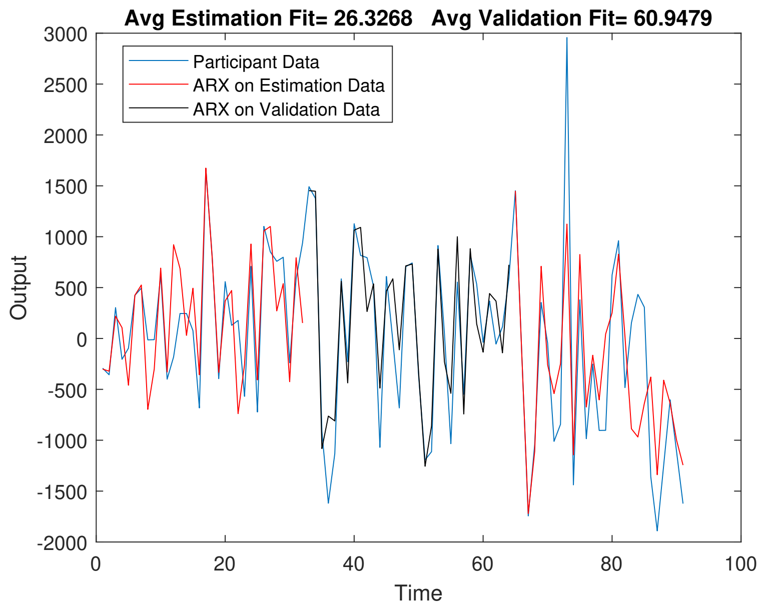

Figure D.2: Participant 008: ARX Model with Feature Select on Estimation and Validation Data for Set #18

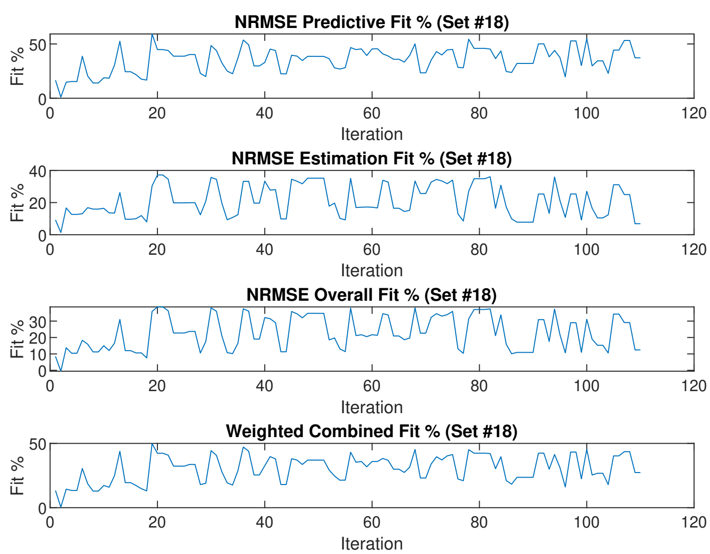

Figure D.3: Participant 008: Fit (%) Iterations for ARX Models with Feature Select and n_k search, Set #18

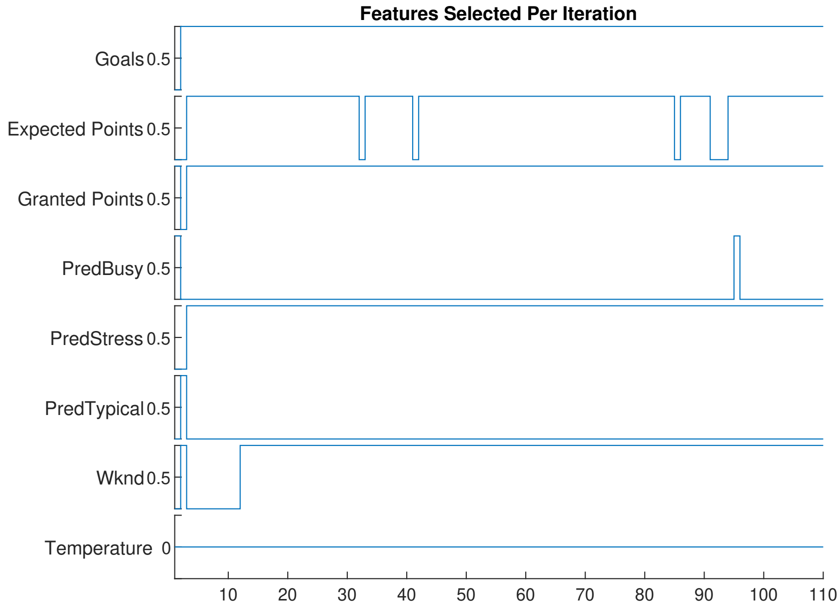

Figure D.4: Participant 008: Feature Select Iterations for ARX Models with Feature Select and n_k search, Set #18

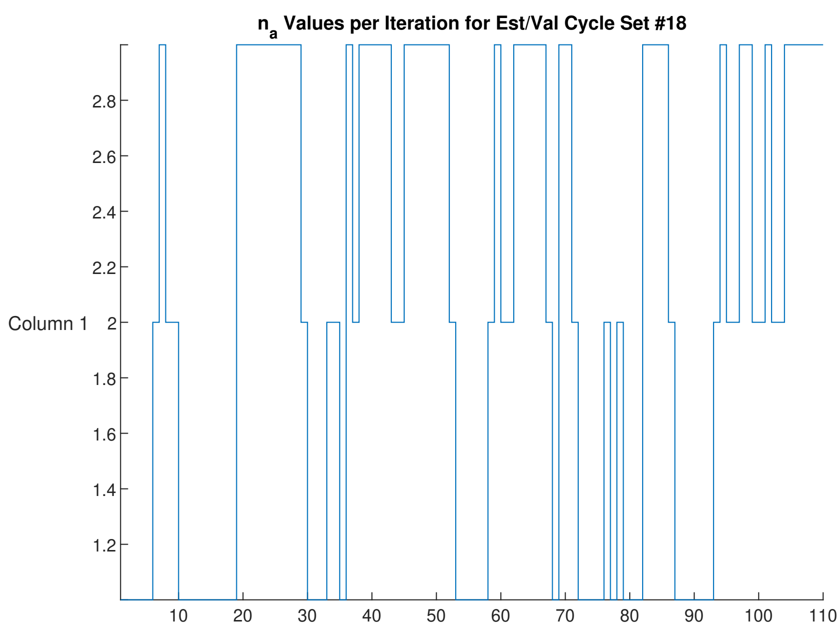

Figure D.5: Participant 008: n_a Iterations for ARX Models with Feature Select and n_k search, Set #18

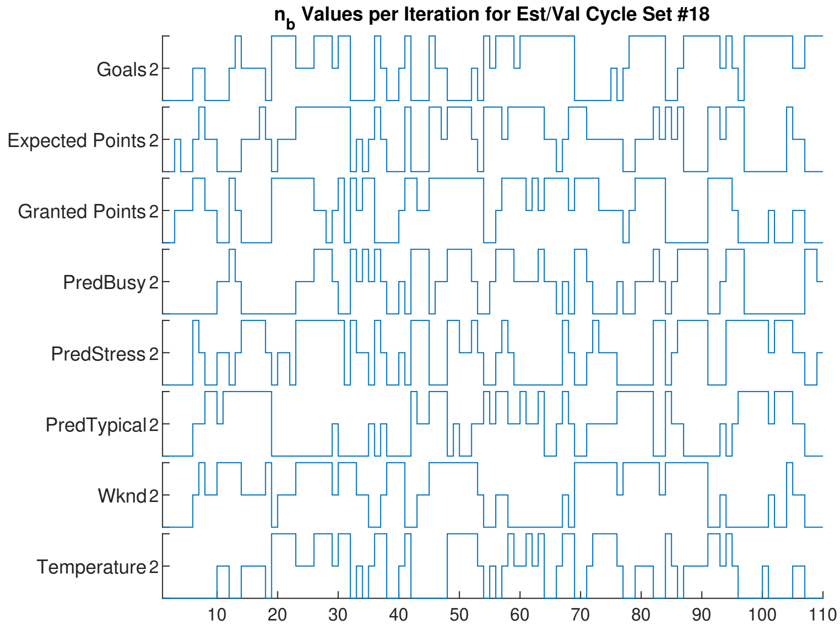

Figure D.6: Participant 008: n_b Iterations for ARX Models with Feature Select and n_k search, Set #18

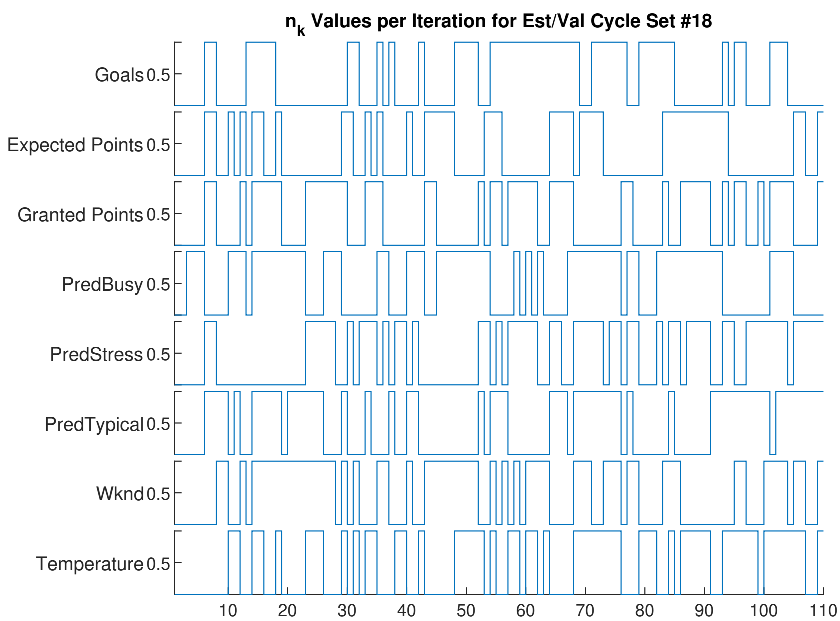

Figure D.7: Participant 008: n_k Iterations for ARX Models with Feature Select and n_k search, Set #18

APPENDIX E

ARX ESTIMATION WITH DSPSA: PARTICIPANT 057 FULL SEARCH ADDITIONAL PLOTS

The following plots are the DSPSA results for set #19 from Figure 4.14 and 4.15, which had the highest weighted fit. To reiterate, the inputs of this model are outlined in Table E.1.

Initialized Features	Selected Features	n_b	n_k
Goals	Goals	2	0
Expected Points	-	-	0
Granted Points	Granted Points	3	0
Predicted Busyness	Predicted Busyness	1	1
Predicted Stress	-	-	-
Predicted Typical	-	-	-
Weekend	Weekend	3	0
Temperature	Temperature	2	0

Table E.1: Participant 057: Feature Selection and n_b, n_k orders ($n_a = 2$)

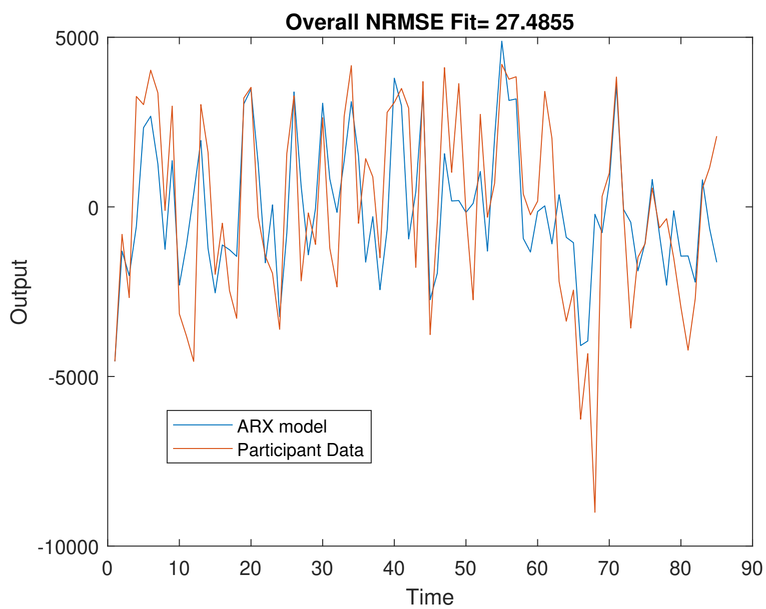

Figure E.1: Participant 057: ARX Model with Feature Select on Overall Data for Set #13

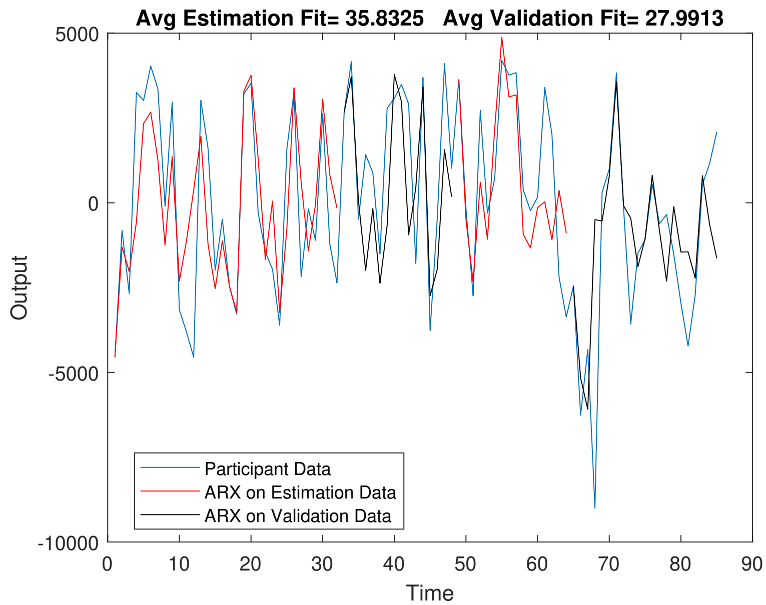

Figure E.2: Participant 057: ARX Model with Feature Select on Estimation and Validation Data for Set #13

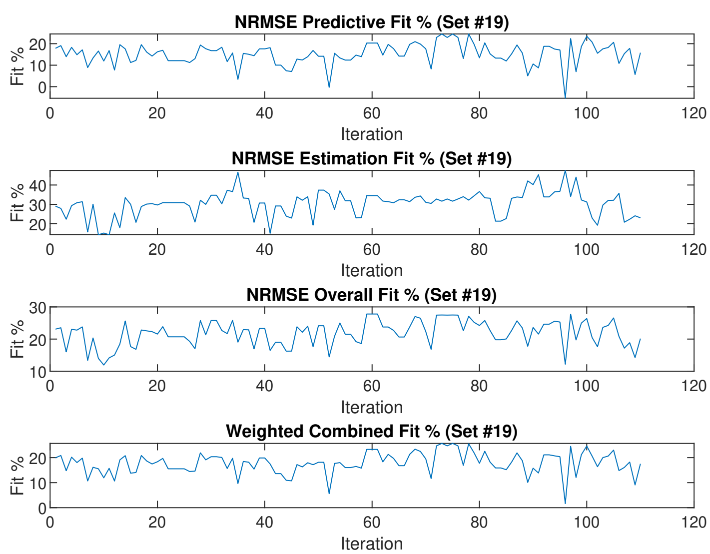

Figure E.3: Participant 057: Fit (%) Iterations for ARX Models with Feature Select and n_k search, Set #13

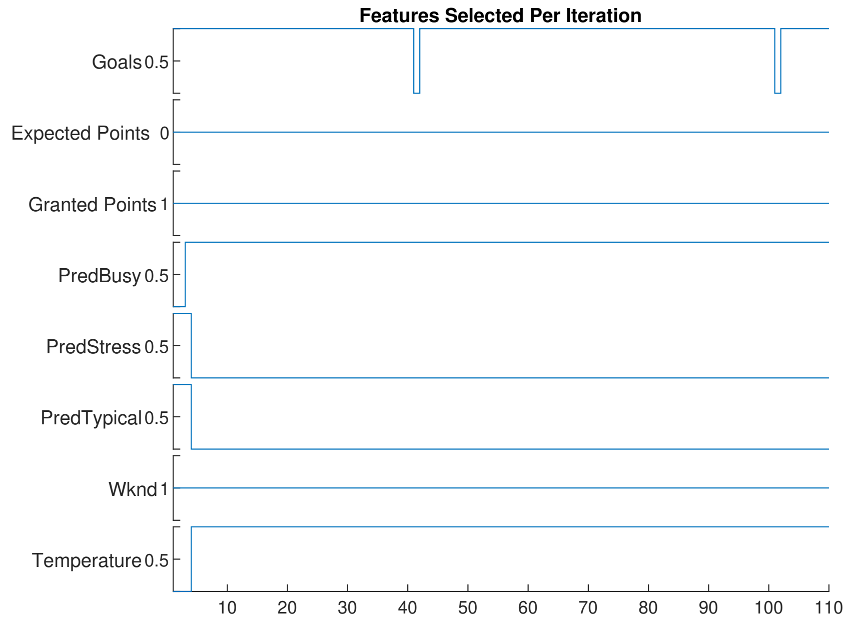

Figure E.4: Participant 057: Feature Select Iterations for ARX Models with Feature Select and n_k search, Set #13

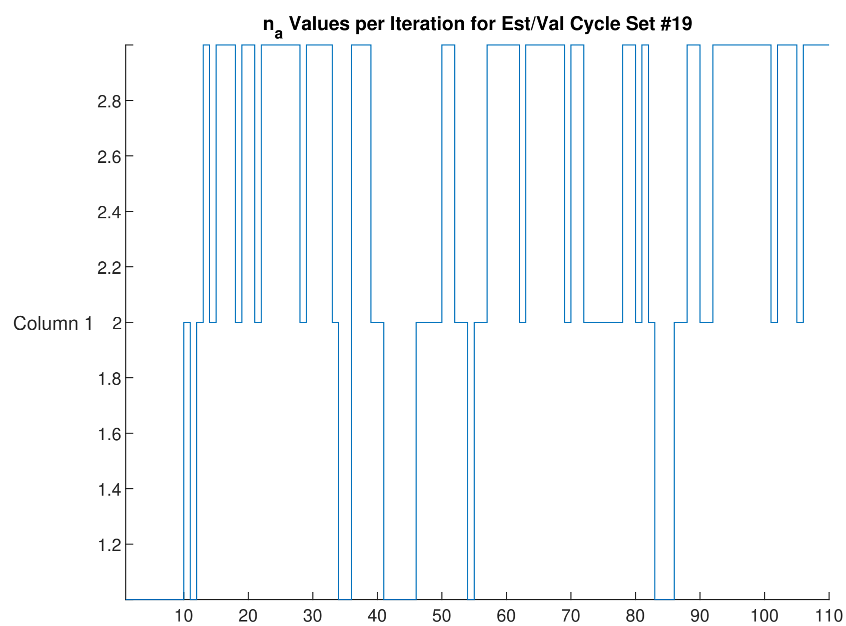

Figure E.5: Participant 057: n_a Iterations for ARX Models with Feature Select and n_k search, Set #13

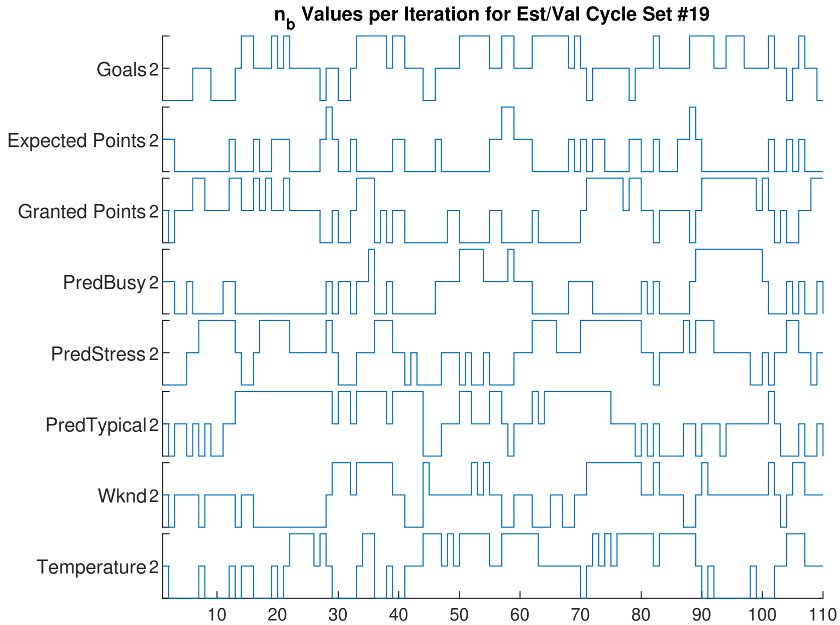

Figure E.6: Participant 057: n_b Iterations for ARX Models with Feature Select and n_k search, Set #13

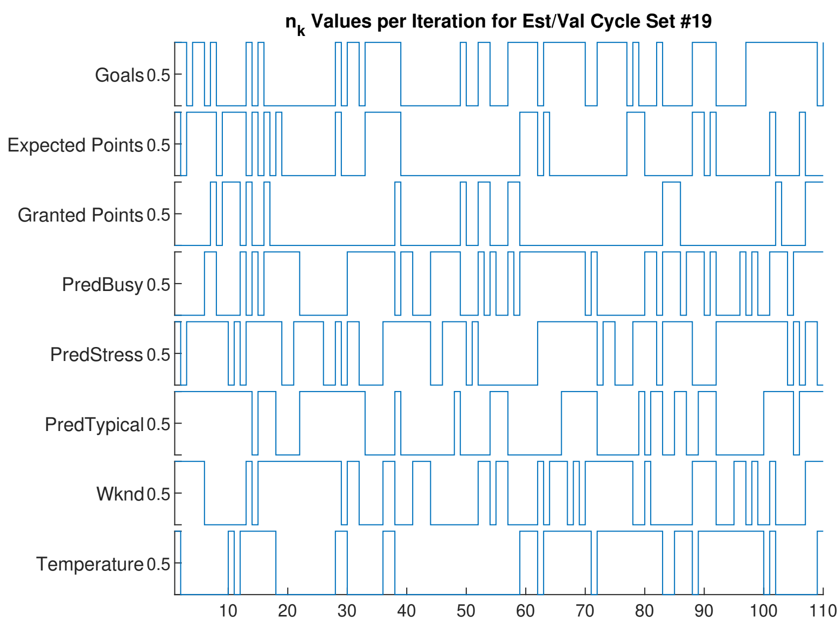

Figure E.7: Participant 057: n_k Iterations for ARX Models with Feature Select and n_k search, Set #13