Adaptive Control of Multistage Airport Departure Planning Process using Approximate Dynamic Programming

Rajesh Ganesan and Lance Sherry

February 15, 2007

Abstract

Many service enterprise systems such as the airport departure systems are typical multistage multivariable systems with non-linear complex interactions between stages. These systems function over a wide range of operating conditions and are subject to random disturbances, which further enhance the non-linear characteristics. Also, there are many uncertain factors which often makes it is difficult to describe the process dynamics with complete information and accurate physical and empirical models. Adaptive controllers based on the analytical and/or artificial intelligence techniques can provide improved dynamic performance of the multistage process by allowing the parameters of the controller to adjust as the operating conditions change, and are known to operate in model free environment. One such example of an adaptive controller, is the combination of analytical dynamic programming methods and artificial intelligence techniques to achieve superior control of operations and improved quality of finished products. This new branch of research has become known as Approximate Dynamic Programming (ADP) methods. This paper first presents a state-of-the-art review including the advantages and limitations of ADP methods. Next, it develops a novel multiresolution assisted reinforcement learning controller (MARLC) based on ADP principles, which is used in an agent-based control model for improving the performance quality of the multistage airport departure planning process. The research is ongoing in collaboration with the Center for Air Transportation Systems Research at George Mason University.

Keywords: Adaptive control, quality improvement, approximate dynamic programming, reinforcement learning, multiresolution analysis, wavelet, national airspace system.

[1]Ganesan is with the Department of Systems Engineering and Operations Research, George Mason University, Fairfax, VA, 22030 (e-mail: rganesan@gmu.edu)

[2]Sherry is the Executive Director of Center for Air Transportation Systems Research, George Mason University, Fairfax, VA, 22030 (e-mail: lsherry@gmu.edu)

1 Introduction

Efficient real-time control of large-scale service, manufacturing, and distribution systems is a challenging task due to the dimension of the problem and the dynamic uncertainties that affect them. Such large systems seldom have accurate process models and are typically made up of interacting smaller subsystems. The conventional linear or fixed parameter controllers are good for a localized control of single or simple process within a given range of inputs, but the same quality of performance under all conditions cannot be maintained. Some of these smaller subsystems if linear can be controlled optimally. However, real world applications are far from being linear and are corrupted with multiscale (multiple features in time and frequency) noise. The control of such non-linear processes corrupted with Gaussian noise (typical assumption) is achieved either through robust or adaptive control (both feedback and feed-forward). They have a centralized architecture, which is often hierarchical for large-scale systems. These centralized control methods have certain limitations which include lack of scalability to an enterprise level due to the large number of variables at that level. Hence, they are only tested to work for single processes or simple distribution systems with a few controllable variables under the assumption that the number of system transition states is finite. In reality, large-scale systems are comprised of a set of interactive hybrid dynamical systems (HDS), consisting of many discrete and continuous variables, which results in a large number of system states (state space explosion). Another limitation arises at the time of practical implementation, when linear assumptions and simplifications on the process models are made because the non-linear complex dynamic models lack the speed of evaluation in real-time. Such simplifications fail to completely capture risks and uncertainties in the system, and the failure to remove multiscale noise further results in sub-optimal control strategies due to which, the efficiency of hierarchical control is restricted. Yet another limitation of conventional non-intelligent control methods includes the lack of an autonomous upset recovery or an auto-reconfigurability feature when the enterprise is subjected to unexpected disturbances. Additional features of multistage systems such as cascading variations in quality characteristics between stages, and scalability to larger systems further motivate the need to research for controllers that can handle such features. The above discussion raises some fundamental questions 1) is it possible to construct and test a general information-driven intelligent controller, which is scalable from an individual process to an enterprise, to control large-scale non-linear systems that operate under uncertainties? 2) can such a control method be adaptive to a hybrid dynamical enterprise system? 3) is it possible to incorporate auto-reconfigurability features, which are enabled by the conversion of information into storable useful knowledge, into such enterprise control methods, and 4) what would be the limitations of the enterprise control system so constructed? Clearly, a better understanding of the concepts of emergence and self-organization are needed, especially from the perspective of designing such

large-scale enterprise systems and synthesis of their interactions. This can be achieved via the design and analysis of intelligent decision support systems (IDSS), which can help enterprises, cope with problems of uncertainty and complexity, and increase their efficiency.

With the above broader perspective of an enterprise control as the final goal of this research, this paper is focused on methods for active control and process adjustment for quality improvement in a multistage system of an airport service enterprise. The purpose of this paper is to review many new schemes that have been proposed recently in the field of adaptive control, motivate the need and present the implementation steps for using approximate dynamic programming (ADP) methods for adaptive control, and discuss the viability of ADP for quality improvement in multistage systems. The paper presents models, solutions techniques, and a real world case study (airport departure planning process) that emphasize the use of an intelligent decision support system for quality improvement in a multistage manufacturing environment. We present the use of wavelet-based multiresolution analysis (WMA) in conjunction with reinforcement learning (RL) to design an intelligent learning-based control approach for multistage systems with multiscale features. In this research we exploit the excellent feature extraction, pattern recognition, data compression, and function approximation capabilities of wavelet analysis, and intertwine them with the RL based controller to obtain a new breed of superior controllers. These are then implemented in a decentralized heterarchical architecture using multi-agents for modeling a multistage airport departure planning system that improves its performance quality. In this architecture, the multistage system is divided into local subsystems, and each agent is associated with a local subsystem that is controlled by the learning-based controller. The decision making capability of the controller is developed using a probabilistic Semi-Markov decision process (SMDP) framework with is solved using ADP techniques. We henceforth refer our learning-based control method as multiresolution assisted reinforcement learning control (MARLC). It is our strong belief that insights presented in this paper will serve as foundation to extend the ADP methods for large-scale enterprise control.

The contributions of this paper are three fold. First, our primary contribution is the novel model-free MARLC methodology, which combines the power of wavelet-based multiresolution analysis and learning-based ADP. Second contribution includes the application of the MARLC methodology to benefit the design and practical implementation of a new decentralized multi-agent control architecture for a multistage airport departure planning system, which is also applicable to other engineering problems-particularly those for which governing equations (predetermined models) are unknown. Final contribution includes the state-of-the-art review of adaptive control methods and their comparison with adaptive control using ADP approaches.

The paper is organized as follows. In Section 2 we review the related literature in adaptive control methods, and provide the motivation for using model-free learning based adaptive control. Section 3

presents the newly developed MARLC methodology that intertwines multiresolution analysis and approximate dynamic programming. In Section 4, implementation of the MARLC for multistage airport departure planning and control is presented, which is followed by conclusions and further research in Section 5.

2 Related Literature

A host of enterprise systems ranging from Manufacturing Execution Systems, Intelligent Manufacturing Systems (IMS), Enterprise Resource Planning (ERP), Advanced Planning Systems (APS) and Customer Relationship Management (CRM) [1], aim to facilitate integration of the manufacturing chain within the networked enterprise, in order to control and to manage the customized manufacturing of both goods and services as desired. Among the above, the IMS has received considerable attention in recent years [2]-[4]. Particularly, the multi-agent systems and holonic manufacturing systems (HMS) have been the latest advancement in the IMS area [5], [6] and has shown a promising trend in manufacturing enterprise control. However, even today IMS is faced with a lack of complete modeling framework for industry, and the extension of the concepts to service sector such as air transportation has not been fully researched. This is mainly due to the lack of tools to test and validate the information-interaction-intelligence complexity that exist in large-scale systems [7], [8]. Consequently, the main paradigm shift in future manufacturing and service automation and control is to bridge the gap between the traditional hierarchical approaches-predetermined modeling approach towards more appropriate heterarchical (often hybrid) approaches-emerging modeling approaches, so that the system can be automatically controlled according to system theory and information-intelligence structure. This can be achieved effectively only via a learning-based control system which has the self-organizing capability to perform autonomously. Our paper aims to assist in bridging the above gap by providing a framework for decentralized model-free learning-based control structure for large-scale service systems and validating it with a real world application to predict and control a multistage airport departure planning system. In what follows we describe the more common model-based control approaches and motivate the need for learning-based model-free approaches.

2.1 Why Model-Free Control?

Control theory has its roots in many disciplines with multitude applications. Typically, control theory is classified into optimal, robust, and adaptive control. However, the literature reviewed for this paper pertains to the model-based and model-free classification of adaptive control theory, and provides a historical motivation both for pursuing the model-free approach and for the need to use wavelets in

control.

2.1.1 Model-Based Control

The model-based controllers use two main types of models: differential-algebraic equations and difference equations. The differential-algebraic equation approach has been used for both linear and linear-quadratic optimal control [9], [10] and control of non-linear systems [11], [12]. Robust control for non-linear systems have been addressed by [13], which in turn reduces to finding a solution to the Hamilton-Jacobi-Bellman (HJB) equation. In recent years, linear and non-linear hybrid dynamical system (HDS) have been the focus of research [14] -[18]. The most popular form of control using difference equations is the Run-by-Run (RbR) controller, in which the control laws are obtained from designed experiments and/or regression models. Some of the RbR algorithms include exponential weighted moving average control (EWMA) [19], optimizing adaptive quality control [20], and model predictive R2R control [21]. Comparative studies between the above types of controllers is available in [22] and [23]. Among the above controllers, the EWMA controller has been most extensively researched and widely used to perform RbR control [24] -[35]. Also, model-based simulation techniques have been used for the control of discrete-event dynamic systems (DEDS) lacking closed form solutions [36] -[40].

Limitations: Some of the primary limitations of above model-based controllers include 1) dependence on good process models, 2) control actions are based on the parameters of filtering method, which are often fixed, 3) cannot handle large perturbations of the system because the system is not intelligent, 4) need multiple filtering steps to compensate for drifts and autocorrelation, and 5) impossible to scale up to higher dimension real-world systems due to the lack of large complex models that can capture the whole system dynamics. One of the ways to handle some of the above drawbacks is through adaptive control.

Many types of model based adaptive control techniques are available in the literature [41] -[43]. These are Dual Adaptive control [44] -[46], Model Reference Adaptive Controllers (MRACs), and Model Identification Adaptive Controllers (MIACs). In these controllers the control law is modified because the parameters of system being controlled changes over time. Again these controllers are only effective in the presence of known process models and are subject to linear assumptions. By linearizing control, these methods have been applied to non-linear systems as well. In the next section we review the relatively new model-free adaptive control.

2.1.2 Model-Free Control

Learning-based model-free control systems, though has been in existence, its potential has not been fully explored. The word model-free is often a misnomer since it is understood as a lack of mathematical construction. Typically, these systems use some form of artificial intelligence such as neural networks,

fuzzy-logic rules, and machine learning and have very strong mathematical foundations underlying their construction. These intelligent controllers have been tested on robots and hierarchical manufacturing systems as well. Some of these systems, particularly neural networks and fuzzy-logic rules, though are claimed to be model-free, do contain certain hidden or implicit models, and make certain strong modeling assumptions when it comes to proving the stability of the controller. Some examples of these controllers include [47] -[52]. Hence, data-driven machine-learning-based controllers (such as the newly developed MARLC) are preferred, and they have been shown to be more effective than neural networks and fuzzy-logic based controllers. However, their wide spread use in the industry has been limited due to the lack of comprehensive studies, implementation procedures, and validation tests. The above types of learning-based control can be further classified based on three major learning paradigms. These are supervised learning, unsupervised learning and reinforcement learning (a strand of ADP).

Neural network based control schemes use supervised or unsupervised learning. In a supervised learning, the learner is fed with training data of the form (x_i, y_i) where each input x_i is usually an n-dimensional vector and the output y_i is a scalar. It is assumed that the inputs are from a fixed probability distribution. The aim is to estimate a function f in $y_i = f(x_i)$ so that the y_i can be predicted for new values of x_i. For a successful implementation of neural network using supervised learning, the training data samples must be of good quality without noise. The learning of the weights on the network arcs during training is usually done using the back-propagation algorithm. In unsupervised learning there is no a priori output. The network self organizes the inputs and detects their emergent properties. This is useful in clustering and data compression but not very useful in control where corrective actions based on outputs are desired. The model-free (information-driven) reinforcement learning-based (RL) control, a simulation-based optimization technique, is useful when examples of desired behavior is not available but it is possible to simulate the behavior according to some performance criteria. The main difference from supervised learning is that there is no fixed distribution from which input x is drawn. The learner chooses x values by interaction with the environment. The goal in RL is not to predict y but to find an x^* that optimizes an unknown reward function $R(x)$. The learning comes from long term memory. In what follows we describe the advantages of reinforcement learning-based control.

2.1.3 Why a Reinforcement Learning-Based Model-Free Control?

These RL-based controllers built on strong mathematical foundations of approximate dynamic programming (ADP) are an excellent way to obtain optimal or near-optimal control of many systems. They have certain unique advantages. One of the advantages is their adaptive nature and flexibility in choosing optimal or near-optimal control action from a large action space. Moreover, unlike traditional process controllers, they are capable of performing in the absence of process models and are suitable for large-scale

complex systems [53]. They can also be trained to possess auto-reconfigurability.

Machine learning based controllers use stochastic approximation (SA) methods, which have been proved to be effective for control of non-linear dynamic systems. In this method the controller is constructed using a function approximator (FA). However, it is not possible for a model-free framework to obtain the derivatives necessary to implement standard gradient-based search techniques (such as back-propagation) for estimating the unknown parameters of the FA. Usually such algorithms for control applications rely on well-known finite-difference stochastic approximations (FDSA) to the gradient [54]. The FDSA approach, however, can be very costly in terms of the number of system measurements required, especially in high-dimensional problems for estimating the parameters of the FA vector. This led to the development of simultaneous perturbation stochastic approximation (SPSA) algorithms for FA, which are based only on measurements of the system that operates in closed-loop [55] -[61]. Among the several variants and applications of SPSA, the implementation of SPSA in simulation-based optimization using RL offers several advantages in solving many stochastic dynamic sequential decision-making problems of which the control problem is a subset [62] and [63]. RL (a strand of ADP) is a method for solving Markov decision processes (MDP), which is rooted in the Bellman [64] equation, and uses the principle of stochastic approximation (e.g. Robbins-Monro method [65]). Howard [66] first showed how the optimal policy for a MDP may be obtained by iteratively solving the linear system of Bellman equations. Textbook treatment of this topic can be found in [67] and [68]. Convergent average reward RL algorithms can be found in [69]. The connection between various control theories and ADP is available in [70] and [71]. Some applications of ADP include electric power market design [72], improved fuel flexibility and efficiency for cars and trucks [73], aircraft control [74], semiconductor manufacturing [75], financial decision making, and large-scale logistics problem [76].

Limitations: Some of the factors that can limit the efficacy to real-time implementation and wider reach of the data-driven simulation-based optimization using RL include 1) data uncertainties (multiscale noise), and 2) 'curse of dimensionality' which prevents scalability due to the storage of large volumes of data. We address these limitations in this research paper.

2.2 Motivation for Wavelets in Control

The wavelet methods, unlike Fourier transform methods, provide excellent time-frequency localized information, i.e. they analyze time and frequency localized features of the system data simultaneously with high resolution. They also possess the unique capability of representing long signals in relatively few wavelet coefficients (data compression). A thorough review of control literature reveals that the potential impact of the use of WMA in control of complex hybrid dynamical enterprise systems remains largely unexplored. We have performed some basic research on the use of wavelet as a denoising tool in

non-linear process control and obtained unprecedented performance improvement, which are summarized in [77] and [78]. The only other work that uses wavelet in process control (again for denoising purpose only) is available in [79]. Due to the exceptional properties of wavelets, we have used WMA as an integral part in several areas of the controller design such as denoising of multiscale random noise of the input data reacting to which reduces controller efficiency, extraction of patterns and data features owing to assignable causes (events, trends, shifts, and spikes) for which the controller must take compensating actions, value function approximation (through diffusion wavelets) that will improve the scalability of the learning component of the controller algorithm, and perhaps result in a higher convergence rate.

3 The MARLC Methodology

This research is primarily driven by the need for a technological breakthrough in real-time adaptive control of large-scale service, manufacturing and distribution systems. The research also addresses the need to find methods that will significantly improve the learning process in ADP, and make ADP suitable for control of large-scale systems. As a step forward in fulfilling the above needs, we present the design of MARLC method, which is applied in modeling a multi-agent control framework for improving the performance quality of multistage processes. The dynamic multi-stage process is assumed to be hybrid with both discrete and discrete abstractions of continuous process variables.

3.1 WRL-RbR Control

A simpler version of the MARLC method was successfully developed (which we named Wavelet-based Reinforcement Learning Run-by-Run Controller, WRL-RbR), and tested it to efficiently control a nanoscale chemical mechanical planarization (CMP) process of silicon wafer polishing. Both model-free and a linearized (for simplicity) model-based version of control were tested on the same process, and compared to non-wavelet and non-learning based approaches. Results from the model-based WRL-RbR version is available in [77]. The purpose of the wavelet analysis in WRL-RbR control was only denoising the system output before it was used for uncertainty prediction and control decision making. This resulted in many benefits which include 1) lower mean square deviation of process outputs, 2) quicker convergence of the expected value of the process to target, 3) faster learning of control actions by the RL algorithm, and 4) protection of the controller against sudden spikes in the noisy process output.

Figure 1a shows a schematic of the model-based WRL-RbR controller. The controller consists of four elements: the wavelet modulator, process model, offset predictor, and recipe generator. The noisy dynamic process output signal $y(t)$ is first wavelet decomposed, thresholded and reconstructed to extract the significant features of the signal. This step eliminates the multiscale stationary noise for which

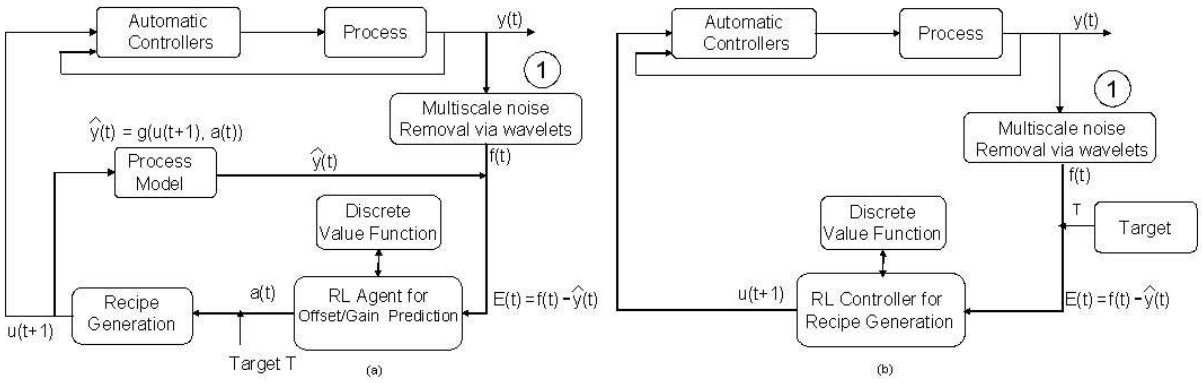

Figure 1: Structure of the (a) model-based and (b) model-free WRL-RbR controller.

the controller need not compensate. The second step involves forecast error $a(t)$ prediction which is accomplished via the RL based stochastic approximation scheme. The input to this step is $E(t) = f(t) - \hat{y}(t)$, where $f(t)$ is the wavelet reconstructed signal and $\hat{y}(t)$ is the predicted model output for the run t. For the CMP process, a simplified linear regression model was chosen as the predicted model (control law). Finally, a control recipe $u(t+1)$ is generated using the forecast error prediction $a(t)$ and gain of the controller b, which is then passed on as set-points to the automatic controller, and to the process model for predicting the next process output at run $t+1$.

Figure 1b shows a schematic of the model-free WRL-RbR controller. The dynamic system is considered as an emergent system, and adaptive control laws are learnt dynamically based on system measurements only. The input to the RL controller is $E(t) = f(t) - T$, where $f(t)$ is the wavelet reconstructed signal and T is the target for the run t. The output of the RL controller is the control recipe $u(t+1)$, which is then passed on as set-point for the automatic controller. Next, we briefly describe the salient features of the WRL-RbR controller.

3.1.1 Salient Features of WRL-RbR Controller

1. The wavelet decomposition was performed using Daubechies [80] fourth order wavelet, and the coefficients were thresholded using Donoho's universal threshold rule [81]. Reconstruction of the signal $f(t)$ in the time domain from the thresholded wavelet coefficients were achieved through inverse wavelet transforms.
2. The evolution of error $E(t) = f(t) - \hat{y}(t)$, (a random variable) during the process runs was modeled as a Markov chain. The process of making forecast error ($a(t)$) prediction decision after each process run was modeled as a Markov decision process (MDP). The MDP model was solved using average reward RL schemes (R-learning).
3. The learning scheme that we have adopted for the WRL-RbR controller was a two-time scale

approximation scheme [69].

4. Once learning was completed, the R-values provide the optimal action choice for each state. At any run t, as the process enters a state, the action corresponding to the lowest non-zero absolute R-value would indicate the predicted forecast error $a(t)$. This was used in the calculation of the recipe $u(t+1)$ using the control law obtained from the process model.
5. For the model-free approach, the RL controller directly learnt the recipe $u(t+1)$ and no process model was used to generate the recipe.

Further details of the WRL-RbR controller is available in [77]. The successful implementation of WRL-RbR served as a motivation for designing the MARLC controller for dynamic enterprise systems, which is presented next.

3.2 Designing the MARLC Architecture for Dynamic Enterprise Systems

Traditionally, in modeling dynamic systems, the hybrid nature has been modeled as purely discrete or continuous systems, which is referred in the literature as aggregation or continuation paradigms [82]. In aggregation, the entire system is considered as a finite automaton or discrete-event dynamic system (DEDS). This is usually accomplished by partitioning the continuous state space and considering only the aggregated dynamics between the partitions. In the continuation paradigm, the whole system is treated as a differential equation. This is accomplished by 1) embedding the discrete actions in non-linear ordinary differential equations (ODE's) or 2) treating the discrete actions as disturbances of some (usually linear) differential equation. Unified model-based approaches have also been suggested as given in [82]. However, literature contains little knowledge on model-free approaches to dynamic systems with hybrid variables. The paper fills this void by designing a model-free MARLC approach for such dynamic systems with both discrete and discrete abstractions of continuous process variables that works as follows: first it captures the features of the process outputs through the WMA analysis. Next, the dynamic system is modeled as a semi-Markov process. Finally, the decision-making process of the control problem is modeled as a semi-Markov decision process (SMDP) and is solved using reinforcement learning-based semi-Markov average reward technique (SMART). The unique feature of the MARLC method is that unlike the usual discretization of the state and action space, the states and actions are approximated as functions using diffusion wavelets. Figure 2 shows the MARLC controller for the dynamic system. In what follows we give a description of the elements of model-free MARLC.

3.2.1 WMA for Multiscale Denoising, Feature Extraction, and Pattern Recognition

The goal of this step is to extract true process information for further processing by the controller. In most real world applications, inherent process variations, instead of being white noise with single scale

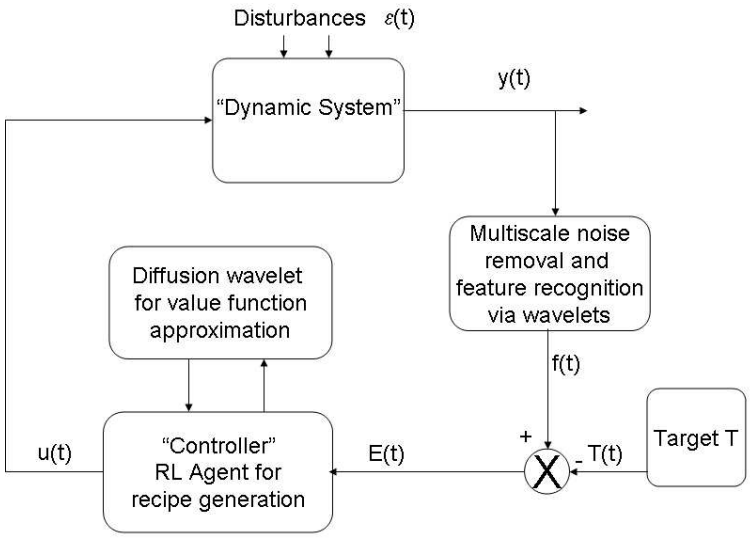

Figure 2: Schematic diagram of the MARLC for Dynamical Systems.

(frequency), are often multiscale with different features localized in time and frequency (example, process industry data on dynamic yield). Thus, the true process outputs $y(\vec{t})$ could be masked by the presence of these multiscale noises. Extracting the true process from a noisy sensor data is critical to take the most appropriate control action. Also, this step prevents the controller from reacting to chance cause of variations in the signal. The task of multiscale denoising and feature extraction is accomplished via the classical WMA method. Other denoising schemes are clearly inefficient as they do not match the excellent localized time-frequency analysis properties of wavelets, and the ability of wavelets to analyze multiple frequency bands at the same time by altering the time window width. Another advantage of wavelet is its ability to analyze data that contain jumps and discontinuities (via translations and dilations of the wavelet), which are characteristics of real-world dynamic systems. Hence, classical WMA is applied on the outputs for multiscale denoising and feature extraction of the dynamic system, which will be accomplished as follows: First, a selection of the best basis function is made using diffusion wavelet (Section 3.2.3). This will help to best represent the original signal with minimal loss of significant features due to wavelet decomposition. Next, wavelet decomposition, thresholding, and reconstruction is performed using the best basis functions via fast wavelet transforms (FWT) to denoise and extract significant features of the signal.

Conceptually, multiscale denoising can be explained using the analogy of nonparametric regression in which a signal $f(t)$ is extracted from a noisy data $y(t)$ as $y(t) = f(t) + noise_1$, where $noise_1$ is the noise removed by the wavelet analysis procedure described below. The benefits of classical wavelet analysis in denoising were noted in the development of WRL-RbR controller. A denoising strategy similar to what was used in WRL-RbR controller will be used for MARLC. After decomposition of the signal into its constituent elements via FWT, denoising will be done using thresholding of the wavelet coefficients $d_{j,k}$

(j is the scale and k is the translation index), which simultaneously extracts the significant coefficients. This will be accomplished by using appropriate thresholding rules. A good review of various thresholding methods and a guideline for choosing the best method is available in [83] and [84]. It is important to select the number of levels of decomposition and the thresholding values in such as way that excessive smoothening of the features of the original signal is prevented. This will be done through the analysis of the energy content in each level of decomposition [85]. Finally, reconstruction of the signal in time domain ($f(t)$) will be done using inverse fast wavelet transforms (IFWT). Pattern recognition will also be done on outputs $y(\vec{t})$ using wavelets to detect unusual events, and to predict trends and drifts in the data. This information along with the denoised outputs is passed on to the RL controller as shown in Fig. 2. A more detailed theory on multiresolution analysis can be found in [86].

3.2.2 RL Based Recipe Generation

The goal of this step is to use the wavelet filtered data and pattern related information to assess the state of the dynamic system and generate (near-) optimal control actions. In Fig 2, the output is $y(\vec{t})$, the disturbance is $\epsilon(\vec{t})$, target $T(\vec{t})$, the permissible control $u(\vec{t})$, and the wavelet-processed outputs are $f(\vec{t})$.

SMDP Model for the Control of Dynamical System

In what follows we provide a brief outline of the SMDP model of the MARLC controller which is solved using RL. Assume that all random variables and processes are defined on the probability space $(\Omega, \mathcal{F}, \mathcal{P})$. The system states at the end of the t^{th} run is defined as the difference between the wavelet processed output $\vec{f}$ and their respective Targets $\vec{T}$, $(E(\vec{t}) = f(\vec{t}) - T(\vec{t}))$. Let $E(\vec{t}) : t = 0, 1, 2, 3...$ be the system state processes. Let $\mathcal{E}$ denote the system state space, *i.e.*, the set of all possible values of $E(\vec{t})$ and $x(\vec{t}) \in \mathcal{E}$ be the system state. In the absence of any event (internal or external), the dynamics are evolving under the influence of disturbances $\epsilon(\vec{t})$ (chance causes of variations). The system is inspected unit time apart, and at every time point t a change in the current action recipes ($u(\vec{t})$) may/may not happen. However, due to randomly occurring events in time (assignable causes of variations), the system is also inspected at those event times and control actions are generated. Thus, the time between inspections of the system follows some general distribution, and the process $E(\vec{t})$ of the dynamic system can be shown to be a semi-Markov process.

Clearly the state transitions in the semi-Markov process of the dynamic system are guided by a decision process that is triggered by events and/or disturbances, where a decision maker selects an action from a finite set of actions at the end of each run. Thus, the combined system state processes and the decision process becomes a semi-Markov decision process (SMDP). Then the control system can be stated as follows. For any given $x(\vec{t}) \in \mathcal{E}$ at run t, there is an action $u(\vec{t})$ selected such that the expected value of the process $y(\vec{t+1})$ at run $t+1$ is maintained at target $T(\vec{t})$. We denote the action space as $u(\vec{t}) \in \mathcal{U}$.

We define reward $r(x(\vec{t}), u(\vec{t}))$ for taking action $u(\vec{t})$ in state $x(\vec{t})$ at any run t that results in a transition to the next state $\vec{x}(t)^{'}$, as the actual error $E(t\vec{+}1) = f(t\vec{+}1) - T(t\vec{+}1)$ resulting from the action. Since the objective of the SMDP is to develop an action strategy that minimizes the actual error, we have adopted average reward as the measure of performance. The SMDP will be solved using SMART [87]. The strategy adopted in SMART is to obtain the R-values, one for each state-action pair. After the learning is complete, the action with the highest (for maximization) or lowest (for minimization) R-value for a state constitutes the optimal action. The learning scheme that we have adopted for the controller is the two-time scale scheme [69]. As in the case of WRL-RbR controller, both the R-values $R(x(\vec{t}), u(\vec{t}))$ and the average reward $\rho(t)$ are learned.

SMART for solving SMDP

Let $\mathcal{E}$ denote the system state space, and $\mathcal{U}$ denote the action space when at the end of the t^{th} run (decision epoch) the system state is $E(\vec{t}) = x(\vec{t}) \in \mathcal{E}$. Bellman's theory of stochastic dynamic programming says that the optimal values for each state-action pair (x, u) (note: now onwards vector notations and time indices have been dropped for simplicity) can be obtained by solving the average reward optimality equation

$$\begin{aligned} R(x,u) = \min_{u\in\mathcal{U}}[\sum_{x^{'}\in E} p(x,u,x^{'})r(x,u,x^{'})] - \rho\delta(x,u) \\ + [\sum_{i\in E} p(x,u,i) \min_{u\in\mathcal{U}} |R(i,u)|] \quad \forall x, \quad \forall u, \end{aligned} \tag{1}$$

where ρ is the optimal average reward, $\delta(x, u)$ is the sojourn time of the SMDP in state (x) under action (u), $p(x, u, x^{'})$ is the transition probability, and $\min_{u\in\mathcal{U}} |R(i, u)|$ indicates that for any state (i), the greedy action (u) for which the non-zero R-value that is closest to zero should be chosen. Value and policy iteration based algorithms are available to solve for the optimal values $R^*(x, u)$ from which optimal policies (u^*) are derived. However, for problems with large state and action spaces, and complicated process dynamics, obtaining the transition probability and the immediate reward matrices are difficult. Even when these matrices are available, carrying out the steps of value and policy iterations could be computationally burdensome. RL based approaches such as SMART to solve SMDP, have been shown to yield optimal values and therefore optimal policies under some conditions.

The learning scheme that we have adopted for MARLC is also a two-time scale scheme [69]. This is because, in this scheme, both the R-values $R(x, u)$ and the average reward ρ (not known apriori) are learned (updated) via the following equations.

$$\begin{aligned} R(t+1)(x,u) \leftarrow (1-\alpha(t))R(t)(x,u) + \alpha(t)[r(x,u,x^{'}) - \rho_t\delta(x,u) + \\ \min_{m\in\mathcal{U}} |R(t)(x^{'},m)|] \quad \forall x, \quad \forall u, \end{aligned} \tag{2}$$

and

$$\rho(t+1) = (1-\beta(t))\rho(t) + \beta(t)\left[\frac{\rho(t)T(t) + r(x,u,x^{'})}{T(t+1)}\right], \tag{3}$$

where $T(t)$ is the cumulative time till the t^{th} decision epoch.

The learning parameters $\alpha(t)$ and $\beta(t)$ are both decayed by the following rule.

$$\alpha(t), \beta(t) = \frac{\alpha_0, \beta_0}{1+z}, \qquad z = \frac{t^2}{K+t}, \tag{4}$$

where K is a very large number. The learning process is continued until the absolute difference between successive R-values in every state is below a predetermined small number $\epsilon > 0$,

$$|R(t+1)(x,u) - R(t)(x,u)| < \epsilon, \qquad \forall x. \tag{5}$$

Once learning is completed, the R-values provide the optimal action for each state (x).

3.2.3 Diffusion Wavelets

The diffusion wavelet method builds basis functions to approximate value functions by analyzing the system state space, which are represented as graphs or manifolds. In the classical wavelet analysis performed on 1-D Euclidean spaces, dilations by powers of 2 and translations by integers are applied to a mother wavelet, to obtain orthonormal wavelet bases. However, for diffusion wavelet, the diffusion operators acting on functions of the state space (and not on the space itself) are used to build the wavelet basis functions. For the MARLC algorithm developed here, the diffusion wavelet helps to obtain the best basis function for multiscale denoising, and for function approximation to mitigate the curse of dimensionality. The details of diffusion wavelets are available in [88], [89] and a brief summary is presented here.

Best Basis Selection

This step helps in obtaining the best basis function for multiscale denoising of the process output $\vec{y(t)}$. This is achieved using diffusion wavelets which generalize classical wavelets. The input to the diffusion algorithm is a weighted graph (G, E, W) and a precision parameter ϵ. The graph can be built from the output data set of a process using Gaussian kernels as follows. For 2 points $x_1, x_2 \in G$,

$$W_{x_1 \sim x_2} = e^{\frac{(\|x_1 - x_2\|)}{\delta}} \tag{6}$$

where δ is the width of the Gaussian kernel. Define D as the diagonal matrix

$$D_{x_1 \sim x_1} = \sum_{x_2 \in G} W_{x_1 \sim x_2}, \tag{7}$$

$$D_{x_1 \sim x_2} = 0 \qquad \forall x_1 \neq x_2. \tag{8}$$

The diffusion operator T is obtained from the Laplacian L of (G, E, W) as

$$T = D^{-0.5} W D^{-0.5} = I - L, \tag{9}$$

where I is an identity matrix. The dyadic powers of T establish the scale j for performing the multiresolution analysis as follows. At level $j = 0$, assume the scaling function $\phi_0 = I$ spans space V_0. The diffusion operator $T^{2^j} = T^1$ for $j = 0$ is obtained from the sparse (QR) factorization of T as in Equation (9). The columns of Q give the orthogonal scaling functions $Q_1 \in V_1$. Using the self-adjoint property of T the next dyadic power T^2 at level $j = 1$ can be obtained from $T^2 = R \times R^*$ where R^* is the complex conjugate of R. Diffusion wavelet basis functions w_1 are obtained via spare factorization of $I - (\phi_1 * \phi_1) = Q_0 R_0$ where w_1 are the columns of Q_0. We select the optimal basis ϕ_j and w_j at level j for a given signal through minimization of the information measure (entropy) [90]. The information measure is defined as a distance measure between the signal and its projection onto the subspace spanned by the wavelet basis in which the signal is to be reconstructed.

Mitigating 'Curse of Dimensionality'

The goal of this step is to make MARLC scalable (handle more variables and higher number of states) by integrating with its learning module, a diffusion wavelet-based function approximation method. For systems with large state-action spaces, the learning algorithms are well known to suffer from 'curse of dimensionality' since they are required to maintain and update an R-value for each state-action combination. One approach to address this computational difficulty is to divide the state-action space with a suitable grid and represent the R-values in each segment of the grid by a function, a method known as value function approximation. In recent years, the concept of diffusion wavelet-based function approximation of state space has been presented to the literature [91]. In this research we have developed fast and stable multiscale algorithms for constructing orthonormal bases for the multiscale approximation of the reward space, and to compute the transform of the reward function in time which is proportional to N, the number of samples in that space. In the case of MARLC, these wavelet basis functions will serve to approximate the reward function by obtaining a multivariate wavelet density estimator as described below.

The multidimensional state and action spaces are represented by monotonically increasing functions $\hat{S} = f_s(x_1, x_2, \cdots, x_{d_1})$ and $\hat{A} = f_a(u_1, u_2, \cdots, u_{d_2})$. Since the range of each state and action variable is known, $\hat{S}$ and $\hat{A}$ are estimated using non-parametric or nonlinear regression methods by sampling from state space $\mathcal{E}$ and action space $\mathcal{U}$ respectively. It is to be noted that this step is executed only once before learning since the range for state and action variables do not change.

Let $\hat{R}$ be the estimated reward density function from $\mathcal{L}_2\mathcal{R}^2$, where $\mathcal{R}^2$ is a 2-dimension state-action space represented using $\hat{S}$ and $\hat{A}$. Mathematically, $R : \mathcal{E}X\mathcal{U} \rightarrow \mathcal{R}$ where R is the reward function and

$\mathcal{R}$ is the R-value space. As the learning algorithm proceeds, $\hat{R}$ is estimated for each segment on the state-action function grid, which is then continuously updated. It is to be noted that the reward function is now a function of estimated state $\hat{S}$ and action $\hat{A}$, which are themselves functions. For a given segment on the state-action function grid, the estimated $\hat{R}(\hat{S}, \hat{A})$ is an image that is obtained using multivariate wavelet density estimation technique as

$$\begin{aligned}\hat{R}(\hat{S}, \hat{A}) = \sum_{k=-\infty}^{\infty} c_{j_0,k}\phi_{j_0,k}(\hat{S}, \hat{A}) + \\ \sum_{j=j_0}^{\infty}\sum_{k=-\infty}^{\infty}\sum_{l=1}^{3} d_{j,k}^{(l)} w_{j,k}^{(l)}(\hat{S}, \hat{A}),\end{aligned} \tag{10}$$

where ϕ and w are obtained using diffusion wavelets as described in the previous section, j is the dilation index, k is the translation index, $l = 1, 2, 3$ are the horizontal, vertical and diagonal wavelet detail index respectively, from 2-dimensional classical wavelet theory. For a given sample of size N from $R(\hat{S}, \hat{A})$ in the state-action function grid, the coefficients can be calculated by

$$c_{j_0,k} = \langle R(\hat{S}, \hat{A}), \phi_{j_0,k}(\hat{S}, \hat{A})\rangle = \frac{1}{N}\sum_{i=1}^{N} R_i(\hat{S}, \hat{A})\phi_{j_0,k}(\hat{S}, \hat{A}), \tag{11}$$

$$d_{j,k}^{(l)} = \langle R(\hat{S}, \hat{A}), w_{j,k}(\hat{S}, \hat{A})\rangle = \frac{1}{N}\sum_{i=1}^{N} R_i(\hat{S}, \hat{A}) w_{j,k}^{(l)}(\hat{S}, \hat{A}). \tag{12}$$

However, fast wavelet transforms (FWT) are used in practice. The coefficients are derived using the cascade (pyramid) algorithm, in which the next level coefficients are derived from the previous level. As new data is generated during learning, only a fixed amount of data is stored in each state-action function grid to obtain ϕ, w, c, and d.

The advantage of the above method is that the reward matrix for each state-action combination is not explicitly stored, which significantly reduces computational memory. The state and actions vectors are stored as functions, which minimizes the need to store explicit values. The wavelet transforms are well known to store information in a compact set of significant coefficients and are excellent methods to compress data in time-frequency domain. This property further helps to minimize computational memory. It is to be noted that even the state and action functions can be obtained using multivariate wavelet density estimation though nonlinear regression was used in this research.

During the learning phase of the MARLC algorithm, ϕ, w, c, and d are updated at every step as $R(\hat{S}, \hat{A})$ values are learnt. In the learnt phase, the estimated state S and $R(\hat{S}, \hat{A})$ are known. An action A is chosen that minimizes $|R(\hat{S}, \hat{A}) - \hat{R}(\hat{S}, \hat{A})|$, which is made possible by the convex properties of the reward functions. The properties of SMART algorithms and assumptions related to learning step size ensure convergence of the algorithm, and that the convergence is to optimal values. The convergent results are available in [69].

4 Implementation of the MARLC for Multistage Airport Departure Planning and Control

In this section we describe the implementation model of the MARLC developed above for performing an agent-based adaptive control of the multistage airport departure planning process. Future trends in manufacturing and service enterprise control, as reported by the International Federation of Automatic Control's (IFAC) coordinating committee [7], consists of high flexibility that allows them to rapidly change to a highly supply and demand networked market, increased automation and control through collaboration, integration and coordination, and a networked decision support system for day-to-day operations with autoreconfigurability feature which is extendable to large-scale complex systems. Agent based approaches have helped in moving away from a centralized hierarchical control structure in manufacturing systems to a more holonic system consisting of autonomous, intelligent, flexible, distributed, cooperative and collaborative agents (holons). In these agent based systems, the efficient synthesizing of vast amount of information to assist the capabilities of decision making under uncertainties and autoreconfigurability cannot be imagined without the use of artificial intelligence techniques such as machine learning. RL-based ADP approaches are a form of machine learning technique, with exceptional capabilities to learn, adapt, and continuously improve the performance of a controllable system. In what follows, we present the background and motivation for our application in multistage airport departure planning, in which a multi-agent based control approach is modeled using MARLC.

4.1 Background and Motivation

This research investigates the most pressing problem of ground delays during the departure planning process at major airport hubs in the National Airspace System (NAS) by modeling a multi-agent based control approach using MARLC technique. The United States NAS is one of the most complex networked systems ever built. The complexity of NAS poses many challenges for its efficient management and control. One of the challenges includes reducing flight delays. Delays propagate throughout NAS and it has a cascading effect. It results in losses for the airlines via cancellations, increased passenger complaints, and difficulty in managing the airline and airport operations since both gate operations and airport air traffic controllers (AATC) could simply be overwhelmed at certain peak hours by excessive demand for take-offs and landings. Delays are caused by several factors, some of which include 1) poor departure planning, 2) near capacity operation of the major hubs, 3) weather, and 4) air traffic management programs such as the ground stop and ground delay program. The total delay of a flight segment from its origin to destination comprises of turn-around time delay, gate-out delay, taxi-out delay, airborne delay, taxi-in delay, and gate-in delay. Among these delay elements, historical data indicates that taxi-out delay contributes to

over 60% of the total delay, which is primarily caused by congestion on ground due to the factors listed above. The taxi-out delay has been increasing over the past years with current averages of about 30-60 minutes after gate push-back at major hubs. Hence, it is imperative to minimize taxi-out delay, which could significantly improve the efficiency of airport operations, and the overall performance of the NAS. As of today, only queuing models, linear-quadratic regression models, and gradient based search methods have been used for taxi-out prediction with marginal success. Such models do not capture the dynamics arising from changing airport conditions and do not provide a feedback mechanism with suggestions to adjust and optimize schedules. This is because of the lack of an Intelligent Decision Support System (IDSS) with a holistic integrated airline-AATC network system that is capable of predicting the airports dynamics under uncertainties and suggesting optimal control measures that could reduce delays. The objective of the MARLC application is to minimize taxi-out delays and achieve optimal traffic flow at airports by modeling and testing a novel machine learning-based IDSS that 1) accurately predicts taxi-out time from a simulated model of the look-ahead airport dynamics, and 2) obtains optimal control actions for airlines and AATC to dynamically adjust departure schedules, make efficient gate, taxiway, and runway assignments, and improve traffic routing.

There is a great potential for increased and efficient utilization of the airport capacity, which is one of the key focus items of Next Generation Air Transportation System (NGATS), as per the report from Joint Program and Development Office (JPDO) [92]. This will also lead to significant improvement in the capabilities for Flow Contingency Management and Tactical Trajectory Management, and will benefit the implementation of an holistic Total Airport Management (TAM) system [93]. As an example of a future concept of automating airport control towers and Terminal Radar Control (TRACON) operations, it will be necessary to predict airport dynamics such as taxi-out times, and feedback this information for aiding artificial intelligence-based decision making at the airport operational level. Improved taxi-out time prediction can be used by airline operating centers (AOC), and airline station operations to increase utilization of ground personnel and resources. Many recent studies have proposed different methods to predict and then use the prediction to minimize taxi-out times. One such study is to predict gate push back times using Departure Enhanced Planning And Runway/Taxiway-Assignment System (DEPARTS) [94], in which the objective for near-term departure scheduling is to minimize the average taxi-out time over the next 10 to 30 minutes, to get flights into the air from the airport as early as possible without causing downstream traffic congestion in the terminal or en route airspace. DEPARTS uses a near-real time airport information management system to provide its key inputs, which it collects from the airport's surface movement advisor (SMT), and recommends optimal runway assignment, taxi clearance and takeoff clearance times for individual departures. The sensitivity of taxi-out delays to gate push back times was also studied using DEPARTS model. Another research that develops a departure

planning tool for departure time prediction is available in [95] -[101]. Direct prediction of taxi-out times has been presented to literature. Such direct prediction methods attempt to minimize taxi-out delays using accurate surface surveillance data [102][101]. One such work is presented in [103], which uses surface surveillance data for developing a bivariate quadratic polynomial regression equation that predicts taxi time. In this work, data from Aircraft Situation Data to Industry (ASDI) and that provided by Northwest Airlines for Detroit DTW (Flight Event Data Store, FEDS) were compared with surface surveillance data to extract gate OUT, wheels OFF, wheels ON, and gate In (OOOI) data for prediction purposes. Algorithms such as space time network search which uses Dijkstra's algorithm, event based A* algorithm, and co-evolution based genetic algorithm have been compared for taxi-time prediction in [104]. Cheng *et al.* [105] studied aircraft taxi performance for enhancing airport surface traffic control in which they consider the surface-traffic problem at major airports, and envision a collaborative traffic and aircraft control environment where a surface traffic automation system will help coordinate surface traffic movements. Specifically, this paper studies the performance potential of high precision taxi toward the realization of such an environment. Also a state-of-the-art nonlinear control system based on feedback linearization is designed for a detailed B-737 aircraft taxi model. Other research that has focused on departure processes and departure runway balancing are available in [106] and [107]. Many statistical models have evolved in recent years which considers the probability distribution of departure delays and aircraft take-off time for taxi-time prediction purposes [108] [109]. For example, queuing models have been developed for taxi time prediction as in [110]. A Bayesian networks approach to predict different segments of flight delay including taxi-out delay has been presented in [111].

With the advent of sophisticated automation techniques and the need to automate airport functions for efficient surface flow movements, the use of information-driven intelligent decision support system (IDSS) to predict and control airport operations has become a necessity. However, industry still lacks the use of intelligent reconfigurable systems that can autonomously sense the state of the airport and respond with dynamic actions continuously. Thus, in many cases decisions are still dependent on human intervention, which are based on local considerations, which are often not optimal. One of the primary reasons for this deficiency is the lack of comprehensive tools for achieving 'automation in decision making', and validated procedures that can simultaneously look at the whole system dynamics, account for uncertainties, and suggest optimal decisions, which can be used by airline and traffic controllers to improve the quality of airport operations. As a first step in the direction of developing such an IDSS for the entire airport, this paper presents a novel MARLC method that uses artificial intelligence to predict taxi-out time, which can be fed back for making optimal schedule adjustments to minimize taxi-delays and congestions. This approach overcomes many limitations of regression model based approaches with constant parameters that are not suitable in the presence of adverse events such as weather that affect airport operations.

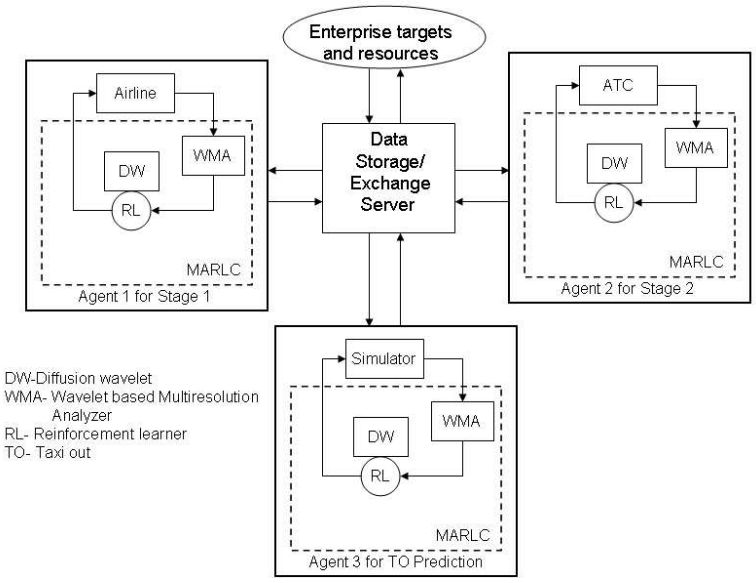

Figure 3: Multi-agent Control Architecture for a Multistage Airport Departure Planning System.

Another limitation arises due to the complex nature of airport operations and the uncertainties involved, which often make it difficult to obtain mathematical models to describe the complete airport dynamics. In such situations model-free learning based techniques can perform better that model based approaches. A unique feature of this model free approach is its adaptive nature to changing dynamics of the airport.

4.2 MARLC Methodology for Multistage Control

The airport departure planning process is a multistage system which can be broadly divided into two stages: stage 1 consisting of airline scheduling and stage 2 is the ATC departure clearance (Fig 3). Each stage is modeled as a locally functioning MARLC-driven agents (1 and 2 respectively) having a SMDP framework. The taxi-out prediction problem is cast in the framework of probabilistic dynamic decision making and is built on the mathematical foundations of dynamic programming and machine learning, which is modeled as agent 3, also having a SMDP framework. The agents interact two-way with a data server. The evolution of airport dynamics (states) is visualized as a sequential dynamic decision making process, involving optimization (actions) at every step. The objective of the departure planning process is to minimize taxi delays, ensure maximum efficiency of airport operation via optimal utilization of resources such as runways, taxiways, gates, and ground personnel, and also maintain safety standards. In what follows the roles played by the agents to improve the performance quality of the multistage airport departure planning system are described.

In Fig 3, agent 3 houses the airport simulator and performs taxi-out prediction. The airport dynamics is simulated using Total Airport and Airspace Modeler (TAAM) available at George Mason University's (GMU) Center for Air Transportation Systems Research (CATSR). The objective of this agent is to

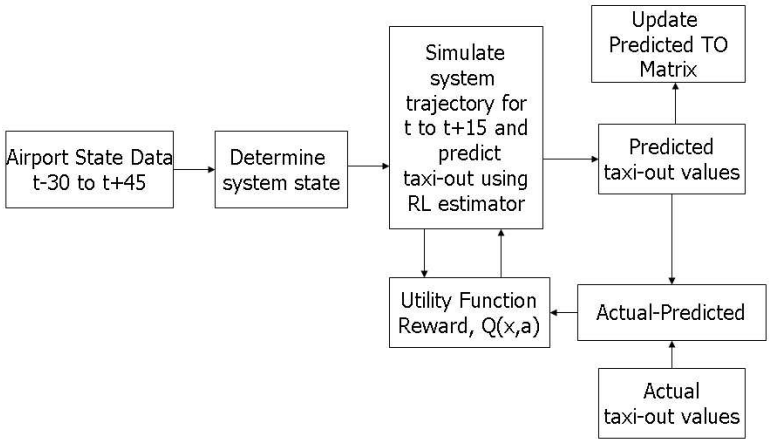

Figure 4: RL based Functional Diagram for Taxi-Out Prediction.

accurately predict taxi-out times. The key inputs to the simulator are airline arrival and departure schedules, airport runway configurations, gate assignments, weather, taxi-way assignments, and airport capacity. This data is obtained from the Aviation System Performance Metrics (ASPM) data base maintained by Federal Aviation Administration (FAA). The RL based functional block diagram for taxi-out prediction is shown in Fig 4. The system state $\vec{x} = (x_1, x_2, x_3, x_4)$ is defined as the number of flights in the departure queue waiting for take off (x_1), number of departing flights taxiing (x_2), number of arriving flights that are taxiing (x_3) and the average taxi time in the last 30 minutes from current time (x_4). A flight is said to be waiting in the departure queue if it has exceeded the nominal taxi time and has still not departed. The purpose of the RL estimator is to predict taxi out time given the dynamic system state. The dynamic system state evolution is modeled as a semi-Markov chain and the prediction process is modeled as a Semi-Markov decision process (SMDP). The SMDP process is solved using RL based stochastic approximation scheme, SMART, as described earlier. The input to RL is the system state and the output of the learning process is a reward function $R(x, u)$ where u is the predicted taxi out values. The utility function (reward) $R(x, u)$ is updated based on the difference between the actual and predicted taxi-out values.

The simulation of airport dynamics is done in a moving window. As seen from Figure 4, the scheduled arrivals and departures up to $t + 45$ minutes is used to obtain the system state $\vec{x}$ where t is the current time. Prediction is done for flights in a moving window of length t to $t + 15$ minutes. This means that for each departing flight in the 15 minute interval from current time, the airport dynamics was simulated for 30 minutes from its scheduled departure time. The window is then moved in 1 minute increments and all flights in the window are predicted again. This means that every flight, unless it leaves before scheduled time, its taxi-out time will be predicted at least 15 times. To calculate average taxi-out times before current time t, actual flight data between t and $t - 30$ are used.

Agent 1 represents the airlines. The purpose of this agent is to dynamically adjust flight schedules within the above moving window time width such that flight departures are evenly distributed especially at peak hours. Today's scheduling issue is that airlines do not collaborate when scheduling flights. This is partly because of competition, government regulations, and also an infeasible task to satisfy every airline with an optimal schedule due to the complexity and size of NAS. Another major reason is the dynamics of the airports that continuously force schedule changes making it impossible to collaborate. Hence, every airline tends to schedule its flights based on its own network connections and passenger demand. This results in multiple flights intending to arrive and depart at the same time due to which both security/gate operations and airport air traffic controllers (AATC) are overwhelmed at certain peak hours. A departure clearance queue forms and AATC handles the requests based on first in first out rules while maintaining safety standards. This situation leads to taxi-out delays arising from congestion. While it is not possible to roll out a schedule that satisfies all airlines and which is robust in the presence of uncertainties, one of the ways to handle the above issue of congestion is to make dynamic adjustments to schedules in a small time window so that departures are evenly distributed. The motivation is that instead of pushing back from gate and waiting 30-40 minutes for take off, a well planned departure schedule that is dynamically adjusted to current airport conditions will reduce the taxi-delay. The insight gained from such adjustments can also be used by airlines to make permanent changes to their departure schedules. We believe that such changes at every major airport that ensures at least a near-optimal departure planning process will improve the overall performance of NAS.

To achieve the above objective of mitigating congestions, agent 1's operation is cast in a MARLC-driven SMDP framework. The input to the agent consists of predicted taxi-out times from agent 3 (indicator of congestion), airline schedules, and OOOI data (gate-out, wheels-off, wheels-on, gate-in) obtained from ASPM database for this research. For a given time window, the system state $\vec{x}$ consists of predicted taxi-out times for individual flights, the aircraft type, and difference between their present scheduled departure times. The action consists of changes to departure schedules within the time window. As learning proceeds, a reward function for the state-action combination is updated. The reward is based on difference between taxi-out times from agent adjusted and non-adjusted schedules. The advantage of this moving window approach is that the schedules are adjusted for only a few flights at a time based on current and predicted (over a 15 minute) airport conditions. Also the model provides for real-time adjustments, which provide the much needed flexibility in the departure planning process.

Agent 2 models the AATC functions. Its objective is to assist the AATC with sequencing departure clearance, taxi-way assignments, and runway assignments. Presently, the AATC operations are human controlled. The operators are trained over an extensive period of time and each operator handles a set number of flights at any given time. This system is prone to human errors such as simultaneous runway

occupancy, near collision situations, and deviations from safe separation distances between departing aircrafts. In order to minimize such human errors a high level of safety standard has been established by keeping larger than required separation distances, but this has resulted in reducing the capacity of the airport. Also sequencing is done manually which is not often optimal. The sequencing must consider time between take-offs which depends on aircraft size in order to avoid wake-vortex effects. Hence, it is imperative to optimize departure sequencing while maintaining safety, which includes optimal sequencing of departure clearance, and gate, taxiway, and runway assignments. Agent 2 will take as input, the departure requests from flights whose times are adjusted by agent 1, taxi-out predictions from agent 3, type of flight, and airport conditions such as weather converted into airport arrival and departure rates. The actions suggested by the RL module of agent 2 are again in a small (15 minute) moving window, which include sequencing departure clearance, taxi-way assignments, and runway assignments. The reward function is updated based on difference between taxi-out times that results from using and not using agent 2's suggestions to the AATC operator. An advantage of using an IDSS for departure sequencing is that it provides optimal action policies for departure clearance, adapts to the changing environment, can help in autoreconfiguration from an unusual event, and also provides higher safety levels for airport operations. This model also paves way for automated AATC operations, which is a futuristic concept.

All of the above agents will be initially trained using simulation-based optimization techniques where each local system is a cast in a SMDP framework and solved using SMART. The inputs (only those that are necessary), the system states (and state space), and the actions (and action space) are defined for each agent. Wherever possible physical laws (if any), and capacity constraints will be fed into the SMDP model to ensure that the actions taken do not violate them. As necessary, wavelet based multiresolution analysis is performed to denoise the inputs and obtain pattern related information about them. Diffusion wavelet is used to obtain the best basis functions for denoising. As learning proceeds, the SMART algorithm within MARLC will use the wavelet-processed data to generate control actions. As the system proceeds from one state to another, the agent's reward matrix will be updated continuously, until optimal actions for each state are learnt. Function approximation methods using diffusion wavelets are integrated with the SMART algorithm for mitigating curse of dimensionality. The learning scheme will also be used to train the system to acquire auto-reconfigurability feature in the presence of adverse conditions. As the system evolves, it is made to learn actions by subjecting it to many extreme situations that are pre-classified as points of reconfiguration at the start of simulation. Once simulation is complete, the implementation is carried out on the real world system using the learnt policies, and further learning for continuous improvement is achieved by using real world data.

It is to be noted that the actual taxi-out time that results from actions taken by agents 1 and 2 are fed

into agent 3 for updating its reward. Thus the system is set up in such a way that every agent attempts to optimize its local system (selfish learning), but at the same time, part of their inputs come from other agent's actions (induces cooperative learning). Also taxi out time is the common performance metric that is used by every agent in its rewarding scheme. A common model evaluation metric is the mean square error (MSE) between the actual and predicted taxi-out values. Also mean, median and standard deviation of the actual and predicted taxi-out times are compared. The RL based estimator is coded using Matlab software.

4.2.1 Benefits of the MARLC methodology

One of the unique benefits of the MARLC method is its ability to effectively handle uncertainties. The change of system state occurs due to events within the local system and also due to events outside the system. These events are both deterministic and stochastic. The multiresolution analysis, an integral part of MARLC (see Fig 2), has the capability to separate noise from the reported system data so that accurate control actions can be initiated. In the absence of any local system model, the MARLC control is a model free adaptive feedback type controller. The idea of a decentralized multi-agent structure provides a framework for intelligent-interactive-information based control where global optimization is an aggregation of several locally optimized interactive subsystems. In general, other benefits such as superior response (often optimal/near optimal) to a given airport condition and an overall improvement in the NAS performance is anticipated from the adoption of the MARLC methodology.

4.2.2 Autoreconfigurability and Scalability

The intelligence capability of a system fitted with MARLC that allows it to get trained, continuously learn, and memorize optimal actions is a unique benefit that is useful at the time of auto-reconfiguration. Another benefit is scalability from individual processes to large enterprise systems. It is to be noted that the enterprise operation is divided into many agents handing only a smaller task by acting on information and using its intelligence. Thus, scalability would mean an increase in the number of agents or sub-systems which is computationally much easier to handle in comparison to a large number of input and controllable variables within a centrally controlled system. Also function approximation methods using diffusion wavelets allow for dimensional scalability by mitigating the curse of dimensionality in RL based ADP systems.

4.2.3 Possible Limitations

The following limitations are anticipated and would be researched in continuation to this research. Real world implementation could be slowed due to the need for interfacing the MARLC method with the

diverse softwares that are already in use. Effect of human intervention is not explicitly studied, however it will be included in future research. In the current model, both agent 1 and 2 adapt to airport dynamics and suggest actions which may be followed or overridden within the local system (airline and AATC) by human inputs. Conflict-resolution and computer-generated negotiations between agents have not been explicitly modeled. For example, the actions taken by agent 2 in departure sequencing are assumed to be acceptable by agent 1 representing airlines. Training labor and managers to adapt and trust the actions taken by the new intelligent MARLC-driven multi-agent approach will be challenging.

4.3 Computational and Validation Studies on the MARLC Method

The scope of this paper is to present the modeling approach for a multiagent learning-based adaptive control that is suitable for multistage system. The taxi-out prediction agent (Agent 3) has been fully developed and tested on Detroit Wayne County Metropolitan Airport (DTW) and the results are available in [112]. Agent 1 and 2 are currently being tested. Comprehensive testing of this method will be accomplished using airport simulation model TAMM in CATSR at GMU. Data for validation is available from the ASPM database maintained by the FAA. A prototype of the IDSS will also be validated on New York, Boston, San Francisco, Atlanta, and Chicago airports, which experience major delays in the NAS.

5 Conclusions

Model-free learning based controllers built using ADP principles though have been in existence, their wide spread use has been limited due to lack of comprehensive studies and tools to implement them. One of the reasons is the curse of dimensionality that results in state and action space explosion. Despite these limitations, it has been argued in recent literature that as the system size and complexity increases, predetermined models are less accurate often due to the lack of complete information (system states are partially observable) and in such cases information (data)-driven learning based systems built on the concepts of dynamic programming and artificial intelligence (called ADP) have been found to be very effective in predicting and controlling such systems. This paper presents a novel control strategy (MARLC), which has high potential in controlling many process applications. The control problem is cast in the framework of probabilistic dynamic decision making problems for which the solution strategy is built on the mathematical foundations of multiresolution analysis, dynamic programming, and machine learning. The wavelet filtering of the process output enhances the quality of the data through denoising and results in extraction of the significant features of the data on which the controllers take action. The paper also presents a new method to overcome the curse of dimensionality using diffusion wavelet based function approximation methods. The MARLC strategy is then customized for a multi-agent based

control model for improving the performance quality of the multistage airport departure planning process. In this model every agent is modeled using MARLC methodology and is trained to improve the quality of a local operation. While the scope of this paper is modeling of the multistage system, our initial results to predict taxi-out time using MARLC have shown promising results, which are not presented in this paper. These performance improvements include an increase in the rate convergence of learning, and a quicker convergence of the expected value of process output on to target, which is due to the intertwining of wavelet analysis with ADP. Further research is underway to comprehensively test the methodology on the complete airport system, and to derive mathematically rational conditions for (near-) optimality, stability and convergence of the MARLC method.

References

[1] A. Ollero, G. Morel, P. Bernus, S. Nof, J. Sasiadek, S. Boverie, H. Erbe, and R. Goodall. Milestone report of the manufacturing and instrumentation coordinating committee: From mems to enterprise systems. *IFAC Annual Reviews in Control*, 26:151–162, 2003.

[2] P. Ed. Valckenaers. Special issue on intelligent manufacturing systems. *Computers in Industry*, 1998.

[3] P. Ed. Valckenaers. Special issue on intelligent manufacturing systems. *Computers in Industry*, 2001.

[4] G. Morel and B. Eds. Grabot. Special issue on intelligent manufacturing systems. *In Engineering applications of artificial intelligence*, 2003.

[5] P. Ed. Valckenaers. Special issue on intelligent manufacturing systems. *Computers in Industry*, 2000.

[6] L. Monostori, B. Kadar, and G. (Eds.) Morel. Intelligent manufacturing systems. In *In Proceedings of the seventh IFAC IMS03 Workshop*, Budapest, Hungary, 2003.

[7] S. Y. Nof, G. Morel, L. Monostori, A. Molina, and F. Filip. From plant and logistics control to multi-enterprise collaboration. *Annual Reviews in Control*, 30:5568, 2006.

[8] G. Morel, H. Panetto, M. Zaremba, and F. Mayer. Manufacturing enterprise control and management system engineering: paradigms and open issues. *Annual Reviews in Control*, 27:199–209, 2003.

[9] W.M. Wonham. *Linear Multivariable Control: A Geometric Approach*. Faller-Verlag, 1979.

[10] A. E. Bryson and Y. C. Ho. *Applied Optimal Control: Optimization, Estimation, and Control*. Hemisphere Publishing Co, 1975.

[11] P. Martin, R. M. Murray, and P. Rouchon. *Flat Systems, Equivalence and Feedback*, pages 5–32. Springer, 2001.

[12] P. Martin, R. M. Murray, and P. Rouchon. *Flat Systems: Open problems, Infinite dimensional extension, Symmetries and catalog*, pages 33–62. Springer, 2001.

[13] J. S. Baras and N. S. Patel. Information state for robust control of set-valued discrete time systems. In *Proc. 34th Conf. Decision and Control (CDC)*, page 2302, 1995.

[14] A. Van der Schaft and H. Schumacher. *An Introduction to Hybrid Dynamical Systems*. Springer, 2000.

[15] A. S. Matveev and A. V. Savkin. *Qualitative Theory of Hybrid Dynamical Systems*. Birkhauser, 2000.

[16] M. Buss, M. Glocker, M. Hardt, O. von Stryk, R. Bulirsch, and G. Schmidt. *Nonlinear Hybrid Dynamical Systems: Modeling, Optimal Control, and Applications*, volume 279, pages 311–336. 2002.

[17] M. Branicky. *General hybrid dynamical systems: Modeling, analysis, and control.*, volume 1066, pages 186–200. 1996.

[18] B. Lennartson, M. Tittus, B. Egardt, and S. Pettersson. Hybrid systems in process control. *IEEE Control Systems Magazine*, 16(5):45–46, 1996.

[19] A. Ingolfsson and E. Sachs. Stability and sensitivity of an ewma controller. *Journal of Quality Technology*, 25(4):271–287, 1993.

[20] E. Del Castillo and J. Yeh. An adaptive optimizing quality controller for linear and nonlinear semiconductor processes. *IEEE Transactions on Semiconductor Manufacturing*, 11(2):285–295, 1998.

[21] W. J. Campbell. *Model Predictive Run-to-Run Control of Chemical Mechanical Planarization*. PhD thesis, University of Texas at Austin, 1999.

[22] Z. Ning, J. R. Moyne, T. Smith, D. Boning, E. D. Castillo, J. Y. Yeh, , and A. Hurwitz. A comparative analysis of run-to-run control algorithms in the semiconductor manufacturing industry. In *Proceedings of the Advanced Semiconductor Manufacturing*, pages 375–381. IEEE/SEMI, 1996.

[23] K. Chamness, G. Cherry, R. Good, and S. J. Qin. Comparison of r2r control algorithms for the cmp with measurement delays. In *Proceedings of the AEC/APC XIII Symposium.*, Banff, Canada, 2001.

[24] E. Sachs, A. Hu, and A. Ingolfsson. Run by run process control: Combining spc and feedback control. *IEEE Trans. Semiconduct. Manufact.*, 8:26–43, 1995.

[25] S. W. Butler and J. A. Stefani. Supervisory run-to-run control of a polysilicon gate etch using *in situ* ellipsometry. *IEEE Trans. on Semiconduc. Manufact.*, 7:193–201, 1994.

[26] E. Del Castillo and A. M. Hurwitz. Run-to-run process control: Literature review and extensions. *Journal of Quality Technology*, 29(2):184–196, 1997.

[27] T. H. Smith and D. S. Boning. Artificial neural network exponentially weighted moving average control for semiconductor processes. *J. Vac. Sci. Technol. A*, 15(3):1377–1384, 1997.

[28] E. Del Castillo and R. Rajagopal. A multivariate double ewma process adjustment scheme for drifting processes. *IIE Transactions*, 34(12):1055–1068, 2002.

[29] R. Rajagopal and E. Del Castillo. An analysis and mimo extension of a double ewma run-to-run controller for non-squared systems. *International Journal of Reliability, Quality and Safety Engineering*, 10(4):417–428, 2003.

[30] S. -K. S. Fan, B. C. Jiang, C. -H. Jen, and C. -C. Wang. SISO run-to-run feedback controller using triple EWMA smoothing foe semiconductor manufacturing processes. *Intl. J. Prod. Res.*, 40(13):3093–3120, 2002.

[31] S. -T. Tseng, A. B. Yeh, F. Tsung, and Y. -Y. Chan. A study of variable EWMA controller. *IEEE Transactions on Semiconductor Manufacturing*, 16(4):633–643, 2003.

[32] N. S. Patel and S. T. Jenkins. Adaptive optimization of run-by-run controllers. *IEEE Transactions on Semiconductor Engineering*, 13(1):97–107, 2000.

[33] C. -T. Su and C. -C. Hsu. On-line tuning of a single ewma controller based on the neural technique. *Intl. J. of Prod. Res.*, 42(11):2163–2178, 2004.

[34] D. Shi and F. Tsung. Modeling and diagnosis of feedback-controlled process using dynamic PCA and neural networks. *Intl. J. of Prod. Res.*, 41(2):365–379, 2003.

[35] E. Del Castillo. Long run transient analysis of a double EWMA feedback controller. *IIE Transactions*, 31:1157–1169, 1999.

[36] C. H. Chen and D. He. Intelligent simulation for alternatives comparison and application to air traffic management. *Journal of Systems Science and Systems Engineering*, 14(1):37–51, 2005.

[37] C. H. Chen. *Very Efficient Simulation for Engineering Design Problem: Modeling and Simulation-Based Life Cycle Engineering*. Spon Press, London, 2002.

[38] Y. C. Ho, R. S. Sreenivas, and P. Vakili. Ordinal optimization of deds. *Journal of Discrete Event Dynamic*, 2(2):61–88, 1992.

[39] W. Gong, Y. Ho, and W. Zhai. Stochastic comparison algorithm for discrete optimization with estimation. *SIAM Journal on Optimization*, 1999.

[40] C. G. Cassandras. *Discrete Event Systems: Modeling and Performance Analysis*. IRWIN, 1993.

[41] K. J. Astrom and B. Wittenmark. *Adaptive Control*. Addison-Wesley, New Jersey, 1994.

[42] S. Sastry and M. Bodson. *Adaptive Control: Stability, Convergence, and Robustness*. Prentice-Hall, New Jersey, 1994.

[43] G. A. Dumont and M. Huzmezan. Concepts, methods and techniques in adaptive control. In *Proceedings of ACC- Transactions on Control Systems Technology*, Anchorage, USA, 2002.

[44] A. A. Feldbaum. Dual control theory. *Automation Remote Control*, 21:874–880, 1960.

[45] N. M. Filatov and H. Unbehauen. Survey of adaptive dual control methods. In *IEEE Proc. Control Theory Appl.*, volume 147, page 118128, 2000.

[46] B. Wittenmark. Adaptive dual control methods: An overview. In *In 5th IFAC Symposium on Adaptive Systems in Control and Signal Processing*, pages 67–72, 1995.

[47] M. S. Ahmed and M. F. Anjum. Neural-net-based self-tuning control of nonlinear plants. *Int. J. Contr*, 66:85–104, 1997.

[48] A. U. Levin and Narendra K. S. Control of nonlinear dynamical systems using neural networks: Controllability and stabilization. *IEEE Trans. Neural Networks*, 4:192–206, 1993.

[49] L. -H. Zhou, P. Han, H. Sun, and Q. Guo. The application of neural network generalized predictive control in power plant. In *Preprints IFAC Symp. Power Plants and Power Systems Control, Seoul, Korea*, page 11511154, 2003.

[50] A. U. Levin and Narendra K. S. Control of nonlinear dynamical systems using neural networkspart ii: Observability, identification, and control. *IEEE Trans. Neural Networks*, 7:30–42, 1996.

[51] K. S. Narendra and K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Trans. Neural Networks*, 1:4–26, 1990.

[52] R. M. Sanner and J. -J. Slotine. Gaussian networks for direct adaptive control. *IEEE Trans. Neural Networks*, 3:837–863, 1992.

[53] O. P. Malik. Amalgamation of adaptive control and ai techniques: Application to generator excitation control. *Annual Reviews in Control*, 28:97–106, 2004.

[54] D. S. Bayard. A forward method for optimal stochastic nonlinear and adaptive control. *IEEE Transactions on Automatic Control*, 36:1046–1053, 1991.

[55] J.C. Spall and J.A. Cristion. Model-free control of nonlinear stochastic systems with discrete-time measurements. *IEEE Transactions on Automatic Control*, 43:1198–1210, 1998.

[56] J.C. Spall. Implementation of the simultaneous perturbation algorithm for stochastic optimization. *IEEE Transactions on Aerospace and Electronic Systems*, 34:817–823, 1998.

[57] J.C. Spall. Adaptive stochastic approximation by the simultaneous perturbation method. *IEEE Transactions on Automatic Control*, 45:1839–1853, 2000.

[58] J. C. Spall. An overview of the simultaneous perturbation method for efficient optimization. Technical report, Johns Hopkins University, MD, USA, 1998.

[59] G. Kothandaraman and M.A. Rotea. Spsa algorithm for parachute parameter estimation. In *AIAA Paper No. 20032118, 17th AIAA Aerodynamic Decelerator Systems Technology Conference and Seminar*, pages 138–148, Monterey, CA, 2003.

[60] H. J. Kushner and G. G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, 2003.

[61] H. -F. Chen. *Stochastic Approximation and its Applications*. Kluwer, 2002.

[62] A. Gosavi. *Simulation Based Optimization: Parametric Optimization Techniques and Reinforcement Learning*. Kluwer Academic, Norwell, MA, 2003.

[63] G. Cauwenberghs. Analog vlsi stochastic perturbative learning architectures. *International Journal of Analog Integrated Circuits and Signal Processing*, 13:195–209, 1997.

[64] R. Bellman. The theory of dynamic programming. *Bull. Amer. Math. Soc.*, 60:503–516, 1954.

[65] H. Robbins and S. Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22:400–407, 1951.

[66] R. Howard. In *Dynamic Programming and Markov Processes*. MIT Press, Cambridge, MA, 1960.

[67] R. Sutton and A. G. Barto. In *Reinforcement Learning*. The MIT Press, Cambridge, MA, 1998.

[68] D. Bertsekas and J. Tsitsiklis. In *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1995.

[69] A. Gosavi. *An Algorithm for Solving Semi-Markov Decision Problem Using Reinforcement Learning: Convergence Analysis and Numerical Results*. PhD thesis, 1998. IMSE Dept., University of South Florida, Tampa, FL.

[70] R. F. Stengel. *Optimal Control and Estimation*. Dover, 1994.

[71] P. Werbos. Stable adaptive control using new critic designs. Technical Report adap-org/9810001, National Science Foundation, Washington D. C., 1998. http://arxiv.org/html/adap-org/9810001.

[72] R. Rajkumar and T. K. Das. A stochastic game approach for modeling wholesale energy bidding in deregulated power markets. *IEEE Transactions on Power Systems*, 19(2):849–856, 2004.

[73] J. Sarangapani and J. Drallmeier. Neuro emission controller for spark ignition engines. In *Cognitive Systems: Human Cognitive Models in System Design*, New Mexico, 2004. Sandia National Laboratories and the University of New Mexico.

[74] D. A. White and D. A. Sofge. *Handbook of Intelligent Control*. Van Nostrand Reinhold, New York, 1992.

[75] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch. *Handbook of learning and Approximate Dynamic Programming*. IEEE Press Series on Computational Intelligence, Piscataway, NJ, 2004.

[76] W. B. Powell, M. T. Towns, and A. Marar. On the value of globally optimal solutions for dynamic routing and scheduling problems. *Transportation Science*, 34(1):50–66, 2000.

[77] R. Ganesan, T. K. Das, and K. Ramachandran. A multiresolution analysis assisted reinforcement learning approach to run-by-run control. *IEEE Transactions on Automation Science and Engineering*, 4(2), 2007.

[78] R. Ganesan. *Process Monitoring and Feedback Control Using Multiresolution Analysis and Machine Learning*. PhD thesis, University of South Florida, 2005.

[79] F. Chaplais, P. Tsiotras, and D. Jung. On-line wavelet denoising with application to the control of a reaction wheel system. In *AIAA Guidance, Navigation, and Control Conference, AIAA Paper 04-5345*, Providence, RI, 2004.

[80] I. Daubechies. *Ten Lectures in Wavelets*. SIAM, Philadelphia, 1992.

[81] D. L. Donoho, I. M. Johnstone, G. Kerkyacharian, and D. Picard. Wavelet shrinkage: Asymptopia? (with discussion). *Journal of the Royal Statistical Society*, 57(2):301–369, 1995.

[82] M. Branicky, V. S. Borkar, and S. K. Mitter. A unified framework for hybrid control: Model and optimal control theory. *IEEE Trans. On Automatic Control*, 43:31–45, 1998.

[83] F. Abramovich and Y. Benjamini. Thresholding of wavelet coefficients as multiple hypothesis testing procedure. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statistics*, pages 5–14. Springer-Verlag, New York, 1995.

[84] M. Neumann and R. V. Sachs. Wavelet thresholding: Beyond the Gaussian iid situation. In A. Antoniadis and G. Oppenheim, editors, *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statistics*, pages 301–329. Springer-Verlag, New York, 1995.

[85] R. Ganesan, T. K. Das, A. K. Sikder, and A. Kumar. Wavelet based identification of delamination of low-k dielectric layers in a copper damascene CMP process. *IEEE Transactions on Semiconductor Manufacturing*, 16(4):677–685, 2003.

[86] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley Cambridge Press, Wellesley MA, 1996.

[87] T. K. Das, A. Gosavi, S. Mahadevan, and N. Marchalleck. Solving semi-markov decision problems using average reward reinforcement learning. *Management Science*, 45(4):560–574, 1999.

[88] J. Bremer, R. Coifman, M. Maggioni, and A. Szlam. Diffusion wavelet packets. Technical Report YALE/DCS/TR-1304, Yale University, 2004. to appear in Appl. Comp. Harm. Anal.

[89] R. Coifman and M. Maggioni. Diffusion wavelets. Technical Report YALE/DCS/TR-1303, Yale University, 2004. to appear in Appl. Comp. Harm. Anal.

[90] R. R. Coifman and M. V. Wickerhauser. Entropy-based algorithms for best basis selection. *IEEE Transactions on Information Theory*, 38:713–718, 1992.

[91] V. Manfredi and S. Mahadevan. Hierarchical reinforcement learning using graphical models. Bonn, Germany, 2005. ICML Workshop on Rich Representation for Reinforcement Learning.

[92] Joint Planning and Development Office. Next generation air transportation system integrated plan. Technical report, url: http://www.jpdo.aero/pdf/ngats-np$_p rogress - report - 2005.pdf, USA$, 2006.

[93] C. Meier and P. Eriksen. Total airport management: A step beyond airport collaborative decision making. Technical report, 2006. http://www.eurocontrol.int/eec/public/standard$_page/EEC_News_2006_3TAM.html$.

[94] W. W. Jr. Cooper, E. A. Cherniavsky, J. S. DeArmon, M. J. Glenn, J. M. Foster, S. C. Mohleji, and F. Z. Zhu. Determination of minimum push-back time predictability needed for near-term departure scheduling using departs. *The MITRE Corporation*, 2001. url:http://www.mitre.org/work/tech$_papers/tech_papers_01/cooper_determination/cooper_determination.pdf$.

[95] J. N. Barrer, G. F. Swetnam, and Weiss W. E. The feasibility study of using computer optimization for airport surface traffic management. Technical Report MTR89W00010, The MITRE Corporation, USA, 1989.

[96] H. R. Idris, B. Delcaire, I. Anagnostakis, W. D. Hall, N. Pujet, E. Feron, R. J. Hansman, J. P. Clarke, and Odoni A. R. Identification of flow constraints and control points in departure operations at airport system. In *Proceedings AIAA Guidance, Navigation and Control Conference*, pages AIAA 98–4291, Boston, MA, 1998.

[97] I. Anagnostakis, H. R. Idris, J. P. Clarke, E. Feron, R. J. Hansman, A. R. Odoni, and W. D. Hall. A conceptual design of a departure planner decision aid. In *Presented at the 3rd USA/Europe Air Traffic Management RD Seminar*, Naples, Italy, 2000.

[98] R. A. Shumsky. Real time forecasts of aircraft departure queues. *Air Traffic Control Quarterly*, 5(4), 1997.

[99] R. A. Shumsky. Predeparture uncertainty and prediction performance in collaborative routing coordination tools. *Journal of Guidance, Control, and Dynamics*, 28(6), 2005.

[100] S. C. Mohleji and N. Tene. Minimizing departure prediction uncertainties for efficient rnp aircraft operations at major airports. In *The MITRE Corporation Center for Advanced Aviation System Development 6th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, pages AIAA 2006–7773, Wichita, Kansas, 2006.

[101] J. Welch, S. Bussolari, and S. Atkins. Using surface surveillance to help reduce taxi delays. In *AIAA Guidance, Navigation Control Conference*, pages AIAA–2001–4360, Montreal, Quebec, 2001.

[102] M. Clow, K. Howard, B. Midwood, and R. Oiesen. Analysis of the benefits of surface data for etms. Technical Report VNTSC-ATMS-04-01, Volpe National Transportation Systems Center, USA, 2004.

[103] D. B. Signor and B. S. Levy. Accurate oooi data: Implications for efficient resource utilization. In *25th Digital Avionics Systems Conference*. Sensis Corporation, 2006.

[104] C. Brinton, J. Kozel, B. Capozzi, and S. Atkins. Improved taxi prediction algorithms for the surface management system. In *AIAA Guidance, Navigation Control Conference*, pages AIAA 2002–4857, Monterey Bay, CA, 2002.

[105] V. H. L. Cheng, V. Sharma, and D. C. Foyle. A study of aircraft taxi performance for enhancing airport surface traffic control. *IEEE Transactions on Intelligent Transportation Systems*, 2(2), 2001.

[106] S. Atkins and D. Walton. Prediction and control of departure runway balancing at dallas fort worth airport. In *Proceedings of the American Control Conference*, Anchorage, AK, 2002.

[107] H. Idris, I. Anagnostakis, B. Delcaire, J. P. Clarke, R. J. Hansman, E. Feron, and A. Odoni. Observations of departure processes at logan airport to support the development of departure planning tools. *Air Traffic Control Quarterly*, 7(4):229–257, 1999.

[108] Y. Tu, M. O. Ball, and J. Wolfgang. Estimating flight departure delay distributions - a statistical approach with long-term trend and short-term pattern. Technical Report RHS 06-034, Robert H. Smith School, 2005. http://ssrn.com/abstract=923628.

[109] R.A. Shumsky. *Dynamic Statistical Models for the Prediction of Aircraft Take-off Times*. PhD thesis, MIT, Cambridge, MA, 1995.

[110] H. Idris, J. P. Clarke, R. Bhuva, and L. Kang. Queuing model for taxi-out time estimation. *Air Traffic Control Quarterly*, 2002.

[111] K. B. Laskey, N. Xu, and C-. H. Chen. Propagation of delays in the national airspace system. Technical report, url:http://ite.gmu.edu/ klaskey/papers/UAI2006$_D elay.pdf, USA$, 2006.

[112] R. Ganesan and L. Sherry. A stochastic dynamic programming approach to taxi-out prediction using reinforcement learning. In *Europe-USA Air Traffic Control Conference*, Barcelona, Spain, 2007.

Rajesh Ganesan received his Ph.D. in Industrial and Management Systems Engineering in 2005 from the University of South Florida, Tampa FL. He is currently an Assistant Professor of Systems Engineering and Operations Research at George Mason University, Fairfax, VA. He served as a Senior Quality Engineer at MICO-BOSCH, India from 1996-2000. His areas of research include air transportation research, wavelet based statistical process monitoring, stochastic control using approximate dynamic programming, and engineering education. He is a senior member of IIE, and a member of INFORMS, and IEEE.

Lance Sherry received his Ph.D. in Industrial and Management Systems Engineering in 1999 from Arizona State University, Tempe, AZ. He is currently the Executive Director for the Center for Air Transportation Systems Research at George Mason University, and as Associate Professor of Systems Engineering and Operations Research. His area of research includes air transportation systems.