2011

Real-time optimal decisions in sequential decision processes with uncertain, exogenous inputs

Yazmin Carroll

REAL-TIME OPTIMAL DECISIONS IN SEQUENTIAL DECISION PROCESSES WITH UNCERTAIN, EXOGENOUS-INPUTS

by

YAZMIN CARROLL

A DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in
The Department of Electrical and Computer Engineering
to
The School of Graduate Studies
of
The University of Alabama in Huntsville

HUNTSVILLE, ALABAMA

2011

In presenting this dissertation in partial fulfillment of the requirements for a doctoral degree from The University of Alabama in Huntsville, I agree that the Library of this University shall make it freely available for inspection. I further agree that permission for extensive copying for scholarly purposes may be granted by my advisor or, in his/her absence, by the Chair of the Department or the Dean of the School of Graduate Studies. It is also understood that due recognition shall be given to me and to The University of Alabama in Huntsville in any scholarly use which may be made of any material in this dissertation.

Yasmin Carroll 5/17/11

(Student signature) (Date)

DISSERTATION APPROVAL FORM

Submitted by Yazmin Carroll in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering and accepted on behalf of the Faculty of the School of Graduate Studies by the dissertation committee.

We, the undersigned members of the Graduate Faculty of The University of Alabama in Huntsville, certify that we have advised and/or supervised the candidate on the work described in this dissertation. We further certify that we have reviewed the dissertation manuscript and approve it in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Electrical Engineering.

______________________ 5-18-11 Committee Chair
(Date)

____

______________________ 5/23/11

Shangbing Ai 6/2/11

______________________ 7/16/11

______________________ 7/21/11 Department Chair

______________________ 07/26/11 College Dean

Rhonda Kay Gaede 11/28/11 Graduate Dean

ABSTRACT

The School of Graduate Studies
The University of Alabama in Huntsville

Degree Doctor of Philosophy College/Dept. Engineering/Electrical and Computer Engineering

Name of Candidate Yazmin Carroll

Title Real-Time Optimal Decisions in Sequential Decision Processes with Uncertain, Exogenous-Inputs

Contemporary decision-theoretic methodologies used to develop "optimal" sequential-decisions for dynamic-processes rely on Markov-models of uncertain exogenous-inputs that act-on the underlying dynamic process. Exogenous-inputs modeled as Markov processes are assumed to satisfy the Ergodic hypothesis and are characterized by long-term statistical characteristics that are somehow known *a priori*. In this dissertation, we consider a special class of uncertain exogenous-inputs with *structured-variations* in their time-behavior and which do not satisfy the Ergodic hypothesis and cannot be effectively characterized in terms of long-term statistical properties. It is demonstrated that a novel alternative to the classical Markov-model, the Structured-Variation modeling technique, may be better suited for calculating optimal sequential-decisions, in that case. It is also shown, through the application of the Principle of Real-Time Optimality, how exogenous-inputs with structured-variations can be effectively incorporated-into the Dynamic Programming method of Bellman where the "state" (exo-state) of the structured-variation type exogenous-input can be estimated by a real-time Kalman filter or state-observer along with the dynamic state of the underlying

sequential decision process. By incorporating the Real-Time Optimality Principle and the Structured-Variation modeling technique in the Dynamic Programming methodology, it is shown that under certain conditions, structured-variation type uncertain exogenous-inputs may be effectively *utilized* in the optimal decision process to achieve more optimal outcomes than could be achieved in the absence of exogenous-inputs. Several specific examples are worked to illustrate incorporation of the proposed alternative model of an uncertain exogenous-input into the Dynamic Programming "solution" of an optimal sequential-decision problem, and how the resulting optimal sequential decisions can, in some cases, result in final outcomes that are more optimal than can be achieved, by optimal decisions, when exogenous-inputs are not present.

Abstract Approval: Committee Chair C Johnson 7-15-11

Department Chair [signature] 7/21/11

Graduate Dean Rhonda Kay Gaede 11/28/11

ACKNOWLEDGMENTS

The work described in this dissertation would not have been possible without the assistance of a number of people who deserve special mention. First, I would like to thank Dr. Carroll Johnson for his suggestion of the research topic and for his guidance throughout all the stages of the work. Second, the other members of my committee have been very helpful with comments and suggestions. Third, I would like to thank the former Dean of Graduate Studies, Dr. Moriarity, for the scholarship that allowed me to finish the requirements for this degree. Fourth, I would like to also thank Jackie Siniard, Senior Staff Assistance at the Electrical and Computer Engineering department, for her invaluable help during all the stages of my graduate work.

Most of all, I would like to thank my family, in particular my children and especially my beloved husband, Michael, who encouraged and supported me patiently during the work on this degree.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

$(.)^T$	Transpose of *(.)*
a	State sequential-decision process scalar coefficient
A	State sequential-decision process coefficient; dimension n by n
$\bar{A}$	Augmented process matrix
a_k	State function coefficient; dimension 1 by n
b	Decision scalar coefficient
B	Sequential-decision process input coefficient; dimension n by r
B_i	Unknown constant
$\bar{B}$	Augmented decision matrix
b_k	Decision function coefficient; dimension 1 by n
C	Unknown constant
$\bar{C}$	Augmented output matrix defined by $\bar{C} = [-C \mid 0]$
$c_j(k)$	Stepwise constant weighting j coefficient in spline representation of exogenous-input
D_k	Decision variable vector in sequential-decision process at time k
D_{RTO}	Real-Time Optimal Decision
D_k^*	Real-Time Optimal decision variable at discrete-time k
$\mathcal{E}$	Effectiveness of Real-Time Optimal Decision D_{RTO}
$E[\,]$	Expected-value operator
F	Exogenous-input matrix; dimension n by p
f_k	Exogenous-input coefficient; dimension 1 by ρ
G	Matrix coefficient in state-vector representation of exogenous-input; dimension ρ by ρ
g	Scalar coefficient in state-vector representation of exogenous-input
$\tilde{G}$	Discretized matrix coefficient in state-vector representation of exogenous-input; dimension ρ by ρ
g_k	Individual accumulated cost at the intermediate discrete stages

f_N	Terminal cost in criterion of optimality
H	Output matrix in state-vector representation of exogenous-input; dimension p by ρ
h	Output scalar coefficient of exogenous-input z
J	Scalar-valued cost function also known as criterion of optimality or performance index
J_{RTO}	Performance index obtained by application of the Real-Time Optimality Principle
k	Discrete-time index (seconds)
K	Constant parameter in geometric description of utility domains
Kx	n by n matrix solution of the first of the set of four coupled difference equations that define P-
Kxz	n by ρ matrix solution of the second of the set of four coupled difference equations that define P-
Kz	ρ by ρ matrix solution of the fourth of the set of four coupled difference equation that define P-
$\mathscr{L}$	Laplace transform
$l_i(k)$	ith basis function in spline representation of exogenous-input
M	Number of basis functions and stepwise constant weighting coefficients in spline representation of exogenous-input
N	Terminal time at the end of the sequential-decision process (seconds0
n	Dimension of state-vector
p	Dimension of exogenous-input vector
$\overline{P}$	Riccati matric difference equation
P_k	Probability distribution
Q	State-weighting matrix in summation term of quadratic performance index J; dimension n by n
R	Decision weighting matrix in summation term of quadratic performance index J; dimension r by r
r	Dimension of decision vector D
s_k	State-evolution function; dimension n by 1
S	Weighting matrix on state at terminal time in quadratic performance index J; dimension $(n+\rho)$ by $(n+\rho)$
$\overline{S}$	Weighting matrix on state at terminal time in the composite quadratic performance index J; dimension $(n+\rho)$ by $(n+\rho)$. Definition: $\overline{S} = \overline{C}^T S \overline{C}$
T	Terminal time at the end of the sequential-decision process (seconds0
t	Time (seconds)

t_0	Initial time (seconds) of the sequential-decision process
$\mathscr{U}$	Utility of exogenous-input (scalar)
$\mathscr{V}$	Minimum value of the performance index J obtained with an Real-Time Optimal decision D_{RTO}
$var(x)$	variance of x
w	Exogenous-Input vector; dimension p by 1
$W(\ )$	z-transform of exogenous-input w
x	Sequential-decision process state-vector; dimension n by 1
$\tilde{x}$	Augmented state vector defined by $col(x_k, z_k)$; dimension $n+\rho$
x_0	Initial value of the state-vector; dimension n by 1
z	State of exogenous-input in sequential-decision process; dimension ρ by 1
$\vec{z}$	Vector representation of exogenous-input state z
ρ	Dimension of exogenous-input vector z
$\sum$	Summation operator
∞	Infinity
λ	Scalar exogenous-input coefficient in coloring-filter
ξ	Time-sequence of independent random numbers
$\bar{\sigma}$	Time-sparse sequence of Kronecker delta functions
σ^2	Variance of exogenous-input w
τ	Sampling period of discretization

Chapter 1

INTRODUCTION

The purpose of this dissertation is to develop an effective alternative to the classical Markov-model in calculating optimal sequential-decisions, when the uncertain exogenous-inputs acting on the underlying system/process have a “structured-variation” type time-behavior. To better understand the motivations for the research presented here, it is useful to review some underlying and related topics.

1.1 Sequential Decision Processes (SDP), Optimality Measures and “Optimal” Sequential Decisions, Bellman’s Principle of Optimality, and the Dynamic Programming Methodology

The determination of smart, effective, real-time sequential-decisions is one of the most common challenges encountered in the high-performance management and operation-of production/distribution/routing processes in industry [13], [24], [47]. A sequential decision process is a procedure requiring a set of choices to be made sequentially for its completion. Each successive decision has to deal with the accumulated consequences of the previous decisions and sometimes with uncertain, unpredictable external circumstances affecting the process. A commonly-used

representation of a decision process model, such as the one presented in Figure1.1 and taken from Puterman [48], consists of decision epochs (or stages), states, actions, rewards, and transformations to go from one stage to the next stage. In the original source of Figure 1.1, the input and output arrows are not labeled and the term "state" is not used in the same context as will be used later in this paper.

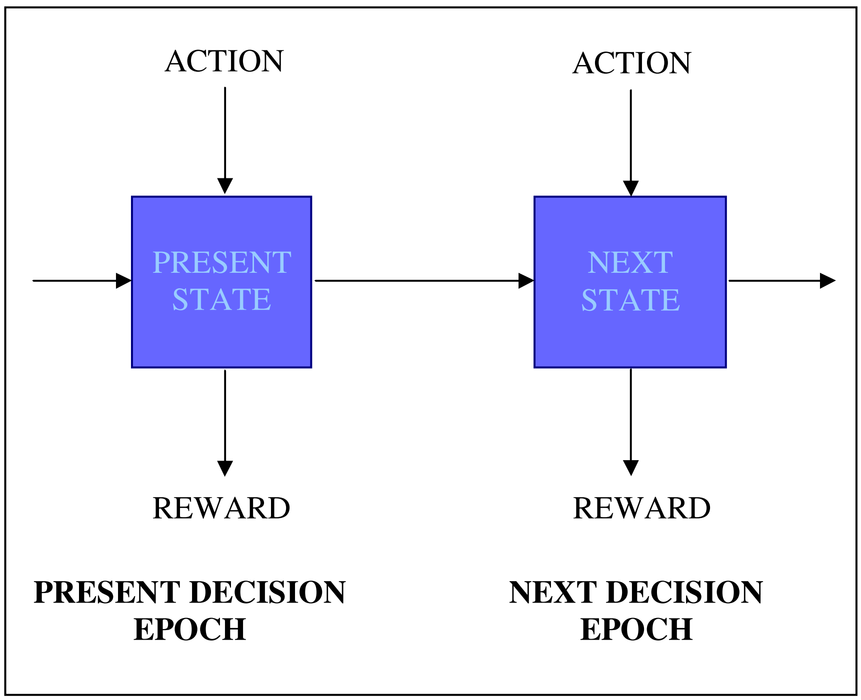

Figure 1.1. Symbolic Representation of a Sequential-Decision Process [48, Figure 1.1.1, page 2]

In the modified representation of a sequential decision process (SDP) shown in Figure 1.2, and introduced by Johnson [38], the concepts of dynamic system-theory are incorporated and attention is focused on the *n*-tuple "process-state" $x \triangleq (x_1, x_2, \ldots, x_n)$ as an *internal attribute* or dynamic "condition" of the process, at each epoch. Thus, the

representation of Figure 1.2 shows that making a decision at one decision-stage, when the underlying process state has the value $x(t_0)$ generates a consequence (=”evolution” of $x(t)$) that determines the state $x(t_1)$ at the *next* decision stage. At each sequential stage, the decision maker must make the “local”, real-time decision D_i, i=1, 2, 3, … based on <u>all</u> the information known at that stage. The effects of each successive decision D_i compound at every stage until the last stage is reached and the “final result” is a “sum” or “convolution” of the compounded decisions. That “final-result” determines the “wisdom” or “optimality” of the collective set of all sequential decisions.

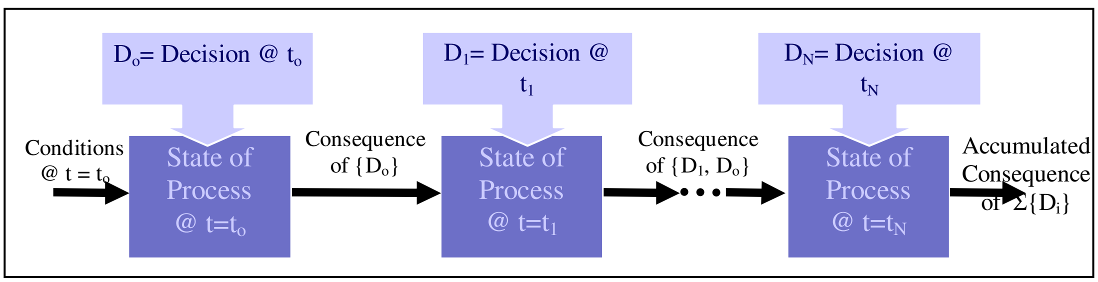

Figure 1.2. Johnson’s Modified Representation of a Sequential Decision Process [38]

The sequential-decision process shown in Figure 1.2 can be applied in many disciplines, for example “optimal allocation of resources” such as money, manpower, energy, air pollution reduction efforts, advertising-funds, production capacity, manufacturing outsourcing, inventory levels, medical procedures, etc., as well as battlefield-resources, troop-deployments, etc. The process model may represent what is actually a continuous time process expressed in discrete time or a process that is inherently discrete-time in nature. The process “boxes” shown in Figure 1.2 can

represent either the *same* process at different times or can represent <u>dissimilar</u> processes at different times or stages. For example, each box can represent the same patient at different times, with the sequential-decisions being which medicines and treatments to give the patient at the decision-times. Or, each box can represent a different process such as in an automobile assembly factory (welding, drive train assembly, painting, testing of electrical systems, etc.).

In order to effectively evaluate the performance of a “decision-maker” in a sequential-decision process quantitatively, it is necessary to choose (a priori) a performance-metric or measure that reflects the underlying goal(s) of the process. Such performance metrics vary with the nature of the process and/or the application specifics and are not necessarily unique. For example, in transportation-related sequential-decision problems, the performance measure might be the minimization of the elapsed-time to transport goods from point A to point B thru a network of rail or roadway connections. On the other hand, the extent of (total) expenditure of resources (gasoline) for the transportation task could be the performance measure required to be minimized. Alternatively, a combination of the two performance measures, minimization of a weighted-combination of elapsed time and total expenditure of resources, could be the required performance-metric to be minimized. In all cases, the objective is to determine the (a) performance measure that best reflects and embodies the desired behavior or outcome from the sequence of decisions.

Thus, choosing a performance measure is equivalent to translating the process’ desired physical requirements into mathematical terms (metrics) that can be scientifically optimized with respect to the sequence of decisions. If the performance measure truly

reflects the desired process performance, then the sequential decisions made in accordance with the scientific optimization procedure become optimal-decisions in the sense that they produce the best possible minimization (or maximization) of that performance measure, under the specified constraints, etc.

It is clear that decisions in SDP-type problems cannot be made in isolation, since the decision-maker must balance the desire for low present “cost” with the undesirability of future “high” costs. The introduction of optimization theories and techniques is a way to capture this tradeoff and optimize the entire sequence of decisions to obtain the “best”-possible outcome. Thus, the term optimization means determining, at each decision-stage, the best possible decision among all feasible alternatives so as to optimize some attribute (performance measure) of the *final* outcome.

Dynamic Programming is a special collection of mathematical tools that can be used to optimize the decisions in sequential decision processes and is commonly attributed to Richard Bellman who presented his work in 1957 [37]. Since then, there has been a constant stream of papers and books which relate Dynamic Programming to practical and theoretical applications involving sequential (multi-stage) decisions.

<u>Remark</u>: Although the Dynamic Programming technique is widely credited to Richard Bellman, little-known information reported in [37] seems to indicate that a co-worker at the Rand Corporation, Dr. Rufus Isaacs, first discovered the essence of the Dynamic Programming idea, in connection with Isaacs “Tenet of Transition”(rule of optimal changes), presented in a series of weekly-seminars at the Rand Corporation and published in a series of RAND Corporation reports during the period 1951-1954, related to a new branch of game-theory Isaacs was then developing called Differential Games

[37]. To avoid unnecessary distractions here, the traditional crediting of Bellman for the Dynamic Programming idea will be followed in this dissertation.

The Dynamic Programming methodology is based on Bellman's Principle of Optimality which asserts that an optimal policy "can be constructed in piecemeal fashion, first constructing an optimal policy for the "tail subproblem" involving only the last stage. Then extending the optimal policy to the "tail subproblem" involving the *last two stages*, and continuing in this manner until an optimal policy for the entire problem is constructed" [16]. In this way and proceeding sequentially "backwards" by first solving the smaller/shorter subproblems, one can, in principle, obtain the true mathematically-optimal policy for the entire process. A simple illustrative example of this "backward" solution process is presented in Chapter 3, see Figure 3.2.

1.2 The Presence and "Adverse" Effects of Uncertain, Uncontrollable Exogenous-Inputs in Sequential Decision Processes

Up to now, we have discussed Sequential Decision Processes without uncertainties. However, uncertainties are a fact of life and are encountered everywhere: changes in gas prices, interest rates, wind gusts, climatic conditions, patient-reactions, etc. These uncertainties, called *exogenous-inputs*, are external influences that are uncertain because the decision-maker cannot know, in real-time, when they will happen or "arrive," and their "nature" or extent is uncontrollable because their arrival and behavior cannot be modified or manipulated by the decision-maker, and they can have a profound effect on the process behavior (such as the result of an exogenous-input acting on an economic system). That unpredictable "effect" can be detrimental-to the goal(s) of the decision process and even negate an otherwise "optimal" decision. This feature is

shown incorporated-into Figure 1.3 which is the same as Figure 1.2, the only difference being that the presence of uncertain exogenous-inputs is now shown as an "input" that directly influences the process behavior at each "stage" and complicates the choice of the best decision. Making a decision at a given stage generates a "consequence" that is now a function of both the decision and the exogenous-inputs acting at that moment and later. The consequences of those "decision-inputs vs. uncertain exogenous -inputs" are compounded until the final stage in which the last outcome (=the ultimate "performance"-metric that embodied the goal(s) of the sequential-decisions) is a complex convolution of all previous decisions and all uncertain exogenous-inputs.

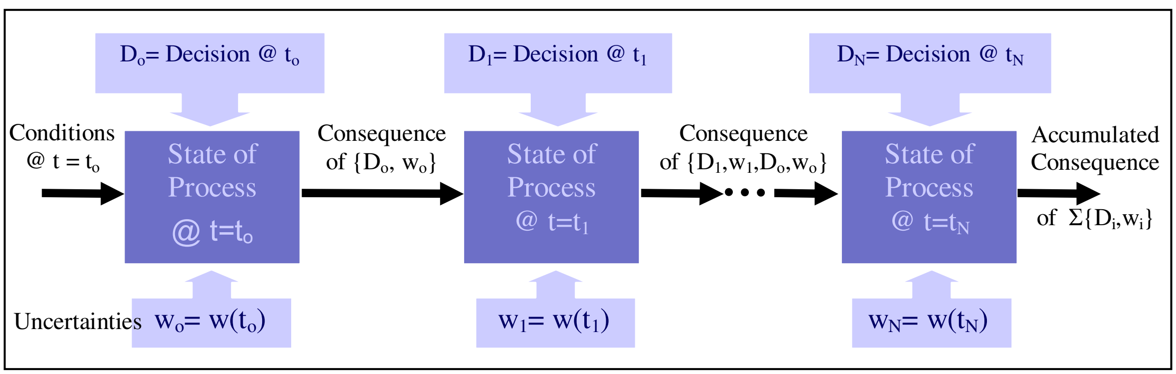

Figure 1.3. Johnson's Representation of a Multistage Sequential-Decision Process with Uncertainties in the Form of Exogenous-Inputs at Each Stage [38]

The traditional method for mathematically modeling the presence of uncertain exogenous-inputs that can act on the underlying process in sequential-decision problems is to represent such exogenous-inputs as Markov-"processes" with *a-priori* "known" statistical/probability characteristics [13], [16]. The validity and effectiveness of the Markov-method for modeling uncertain exogenous-inputs that have a consistent "noisy,

erratic" type characteristic-behavior that can be reliably-modeled in terms of various "random-process metrics" has been established and demonstrated in a wide-variety of practical applications [13], [48], [14]. However, in some sequential-decision problems, the uncertain exogenous-inputs that act on the underlying process are *not* characterized by "noisy, erratic" behavior and *do not* satisfy the theoretical assumptions underlying the Markov-modeling methodology, with the result that the effectiveness of otherwise "smart" sequential-decisions, based-on Markov-models of exogenous-inputs, can be compromised.

1.3 Overview of the Dissertation and Purpose of the Study

In this dissertation we first describe a *particular class* of uncertain exogenous-inputs that do not satisfy the technical assumptions underlying the Markov-modeling procedure traditionally employed in SDP. This class of uncertain exogenous-inputs is known as exogenous-inputs with "structured-variations" [35]. We then employ an alternative modeling procedure as developed in [35] for that particular class of exogenous-inputs which has the benefit of enabling the modeling of such exogenous-inputs by information-rich state-variable modeling and estimation techniques. Using this technique, we validate the assertions in [38] that when exogenous-inputs with structured-variations are incorporated into optimal sequential-decision processes using the Dynamic Programming methodology, the "state" Bellman loosely refers-to in his Principle of Optimality *cannot* be "known" a-priori thereby making it *computationally-impossible* to employ Bellman's "backward-evolution" solution procedure associated with the Dynamic Programming methodology and Bellman's Principle of Optimality. However, by

invoking the more general Principle of Real-Time Optimality (RTO), as proposed in [38], one can introduce a "new" *augmented state-vector* that permits the use of Bellman's backward-solution procedure to derive sequential-decisions that are "optimal" in the RTO sense--- <u>which is the "best" rational</u> <u>(non-gambling) decision strategy one can follow when the exogenous-inputs have structured-variations</u>. This process of modeling uncertain exogenous-inputs using the structured-variation methodology and applying the RTO principle in SDP by Dynamic Programming, leads us to discover that structured-variation type exogenous-inputs, which normally have a negative connotation (adverse-effects) regarding achieving "optimal-outcomes" in SDP problems, can have a surprisingly positive effect or positive "utility" by enhancing the optimality (achieving a <u>higher</u> degree of desired performance) of the decision process if the sequential-decisions D_i are decided in an exceptionally "smart" manner using a novel decision-algorithm or policy as developed here.

In Chapter 2, the mathematical representation of a general class of sequential-decision process and definitions of the concepts of "optimal"-decisions, "state" and Bellman's Principle of Optimality, are presented. A review of the current literature on the solution of multi-stage Sequential-Decision Processes by Dynamic Programming is presented in Chapter 3, where Bellman's Principle of Optimality is explained in detail with the use of examples. The assumptions and characteristic features of the traditional Markov procedure for modeling uncertain exogenous-inputs will also be presented in Chapter 3, along with an introduction to the structured-variation technique for modeling the time-variations of uncertain exogenous-inputs and the new Principle of Real Time Optimality (RTO) as introduced by Johnson in [38]. In Chapter 4, a detailed procedure is

developed to incorporate structured-variation type exogenous-inputs and the Principle of Real Time Optimality *into* the Dynamic Programming methodology, using exact analytical methods for a specific multistage, first-order sequential-decision process and a general class n-stage first-order sequential decision process. In Chapter 5, the concepts of burden, assistance and "utility" of an exogenous-input are presented and defined precisely. Then the positive, negative and zero utility domains for a sequential decision process with structured-variation type exogenous-inputs are defined and derived. The results in Chapters 4 and 5 are illustrated in Chapter 6 by applying them to some specific analytical examples. Finally, a summary of the results obtained in this research, along with conclusions and recommendations for further work are presented in Chapter 7.

Chapter 2

PROBLEM DEFINTION

2.1 Mathematical Model of a Sequential-Decision Process

The basic sequential-decision process considered in this dissertation is a discrete-time dynamic process where the sequential-decisions are made over a finite number of distinct time-stages. The mathematical-model of that class of processes is assumed to have the following generic "state-variable" form [16]

$$x_{k+1} = s_k(x_k, D_k, w_k), \;\; k = 0, \; 1, \; \ldots, \; N-1, \tag{2.1}$$

where k is the discrete-time index, (N-1) is the last decision stage and x_k, D_k, and s_k are defined below. The class of dynamic sequential decision processes defined by (2.1) has the following features, as described by Johnson in [38]:

1. A valid *dynamic* "state" denoted by x_k which is an n-tuple commonly referred-to as an n-dimensional state-vector. The set of axiomatic conditions that comprise a valid dynamic state for use in the Dynamic Programming method of solving SDP's, an important issue not commonly discussed in the Dynamic Programming

literature, are addressed in Johnson's paper titled "Beyond Bellman's Principle of Optimality; The Principle of Real-Time Optimality" [38] and will be discussed in the following subsection.

2. A multivariable (vector) decision-variable denoted by D_k determined at each decision time k (in real-time). The systematic determination of the "best" sequence of decisions $\{D_k\}_0^{N-1}$ is the core of the sequential-decision optimization problem.

3. A collection of uncontrollable, uncertain, unmeasurable exogenous-inputs (disturbances) w_k acting on the process at time k, which form a p-dimensional vector.

4. A n-dimension state-evolution function $s_k(x, D, w; k)$ which is assumed to be a *known* n-vector function that describes how the n-dimension state x of the underlying dynamic process *transitions sequentially* from one stage (k) to the next ($k+1$) and how that transition relates to the value of the process state x, the decision-variables D_k and the exogenous-input w_k at the decision-time.

5. A scalar-valued cost function or measure of performance J whose minimization/maximization embodies the user-specified desired outcome of the sequential-decision process, such as minimization of an undesirable feature of the outcome or maximization of a desirable feature of the outcome. The measure of performance J is commonly referred-to as the "criterion of optimality" and it accumulates at each discrete time k until the end of the sequential-decision process at the "terminal-stage" N. The numeric value of $J(N)$ relates the

optimization measure to the set of decision-variables D_k and the initial-value of the dynamic state x_k. That is, the criterion of optimality *J* generally "ranks" the decisions $\{D_k\}_o^{N-1}$ based on the sum of the present cost and the expected future cost. The cost function *J* describes how the cost incurred at each stage (time) *k* *accumulates* over time, leading-to the grand-total cost, at the "end" of the decision-sequence, expressed by the scalar-function

$$J = g_N(x_N) + \sum_{k=0}^{N-1} g_k(x_k, D_k, w_k)\,, \tag{2.2}$$

where $g_N(x_N)$ is the component of *J* referred-to as the terminal cost incurred at the end of the process (stage *N*), and the indicated sum of the $g_k(x_k,D_k,w_k)$ with k=0, 1, 2, …, (*N*-1) is the component of *J* that reflects the accumulated (individual) costs at the intermediate discrete stages (times) *k* [16].

In the general case, the process state x_k, the decision variable D_k, and the exogenous-input w_k can each be multidimensional. However, to simplify visualization of the concepts presented in this dissertation, they will be restricted to the case in which x_k, D_k, and w_k are scalars.

2.2 The Concept of "Optimal-Decisions" in Sequential-Decision Processes

In a large subset of the literature on sequential-decision processes, the term "state" is loosely used as equivalent to an "input" into the decision process [47] which is in great dissimilarity with the way the term "state" is used in the mathematical theory of

dynamic-processes. As mentioned in the previous subsection, Johnson has addressed [38] the importance and the technical preciseness of the concept of process-state in Bellman's Dynamic Programming Methodology. According to [38], the relevance of Bellman's Principle of Optimality and associated solution methodology relies on the existence and knowledge of a suitable "state" (called a "Bellman-State" by Johnson) for the underlying process which in turn can enable determination of the final value of the optimization measure. Thus, to be suitable, the process "state" must embody all the information necessary to evaluate, at each possible process "initial-condition," the *future* consequences of all the decisions and all the past, present, and future time-behaviors of all the inputs that act-on the process.

The changes in the state of the sequential-decision process can be strongly influenced by unpredictable and unmeasurable exogenous-inputs that act on the process. The effects of such exogenous-inputs may influence the sequence of decisions and thereby affect the final value of the optimization measure J. Since, in general, the future time-variations of the exogenous-inputs are not known *a-priori,* not directly *measurable* nor *reliably-predictable,* their ultimate influence on the final value of the optimization measure J cannot be reliably evaluated until the sequential-decision process has ended. This is a "surprisingly stringent, and often underestimated, technical demand on not only the knowledge" of the state of the process "but also on precise and complete, quantitative knowledge of the entire future time-behavior" of all the exogenous-inputs that may influence the process and in turn the optimization measure [38].

This requirement to know the entire future time-behavior of the exogenous-input is not a matter of concern in the sequential-decision process if the time-behavior of such

exogenous-inputs is accurately known beforehand (such as gravity effects and tide-behaviors) or if the time-behavior of the exogenous-input is uncertain but "noisy" and can be modeled as a sample function of an *Ergodic* random-process with known, reliable statistical metrics (mean, variance, etc.).

However, in some instances, the uncertain exogenous-inputs affecting the sequential decision process are *non-noisy* in nature and have uncertain time-variations that behave in certain qualitative patterns that are knowable. Those exogenous-inputs *cannot* be accurately represented as a conventional random-process. In those cases, the "Bellman state" of the sequential-decision process *is not* a valid state since such a "state" does not embody sufficient information to determine precisely the consequences of past, present, and future time-behavior of all the inputs acting on the process, and specifically the consequences caused by the (then) unknown "future" exogenous-inputs.

A proper "Bellman-State" for a sequential-decision processes with non-noisy, uncertain exogenous-inputs having "structured-variations" will be presented, and incorporated in the Dynamic Programming solution process, in Chapter 4.

Chapter 3

SOLUTION OF MULTI-STAGE DECISION PROCESSES BY DYNAMIC PROGRAMMING

This chapter presents a research of the literature on the dynamic programming approach to sequential decision processes as described by Bellman [9], Howard [28], Blackwell [17], Denardo [23], and Veinott [51] and many others. To explain how Dynamic Programming works, in this chapter we first look at a sequential-decision process that does not have any exogenous-inputs influencing it. We then consider the same decision process with exogenous-inputs acting on the underlying process and explain how these exogenous-inputs are traditionally modeled in the sequential-decision literature.

3.1 Dynamic Programming and the Principle of Optimality

One of the most effective and widely-used methodologies for determining "optimal" sequential decisions that optimize (minimize or maximize) performance criteria of the form (2.2) is the method of Dynamic Programming as introduced by Richard Bellman [4]-[8], [12], [13]. To explain how Dynamic Programming works, we

revisit the multi-stage system of Chapter 1 shown (repeated) here as Figure 3.1. From Figure 3.1, it can be observed that the stages are joined together in series so that the output of one stage becomes the input to the next. The *N*-stage optimization problem is to maximize/minimize the *N*-stage optimization criterion J_N over the variables $D_0,\ D_1,\ \ldots,D_{N-1}$, that is, to find the optimal *J* as a function of the initial process state at time t_0. This is accomplished using Bellman's ingenious backward process in which the starting stage is *N-1*. At this stage, we solve for the optimal decision that gives the maximum/minimum *J* for an arbitrary state x_{N-1}. Then, we move backward one stage to (*N-2*) and once again solve for the optimal decision D_{N-2} that gives the state x_{N-1} which links the decision D_{N-1} to the decision at *(N-2)* and gives the maximum/minimum J_{N-1}.

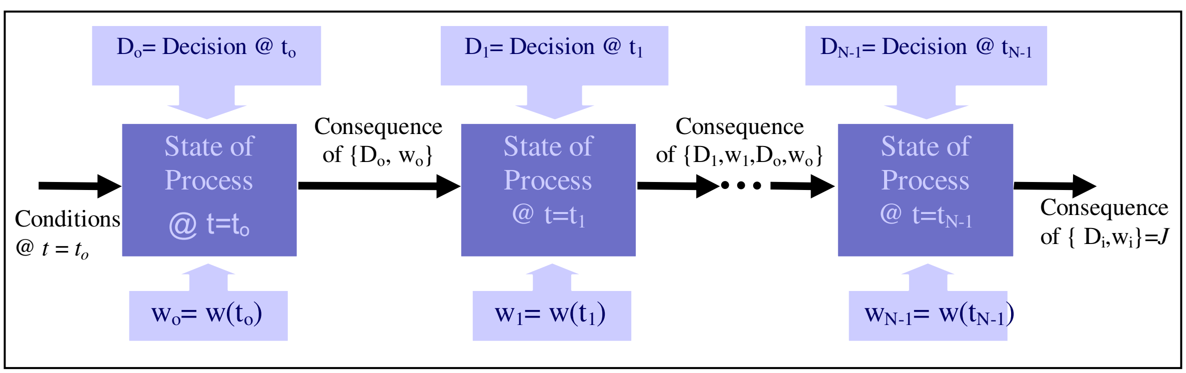

Figure 3.1. Multi-Stage Sequential-Decision Problem [38]

This backward recursive process is repeated until we reach the first stage, at which point we have obtained a set of optimal decisions at each stage *k* in terms of the process state x_k. In this way, we have replaced the multi-stage decision process with a

sequence of *one-stage* decision processes. This process is the basis of Dynamic Programming and is what Bellman called ***The Principle of Optimality*** which states that "an optimal policy has the property that whatever the initial state or decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision" [47].

3.1.1 Dynamic Programming Example 1: The Shortest Route Problem

To easily understand how Dynamic Programming works, we will look at a very simple and typical example that shows the backward technique clearly. It is called the Shortest Route Problem but it is really a quickest-route problem. It was taken from Denardo [24] although other authors have used this example as a first introduction to the Dynamic Programming technique and the Principle of Optimality [44], [15]. Figure 3.2 illustrates this sequential-decision process in a directed network. In this network, the nodes are road junctions represented by the circles and the arcs are the arrows connecting the circles.

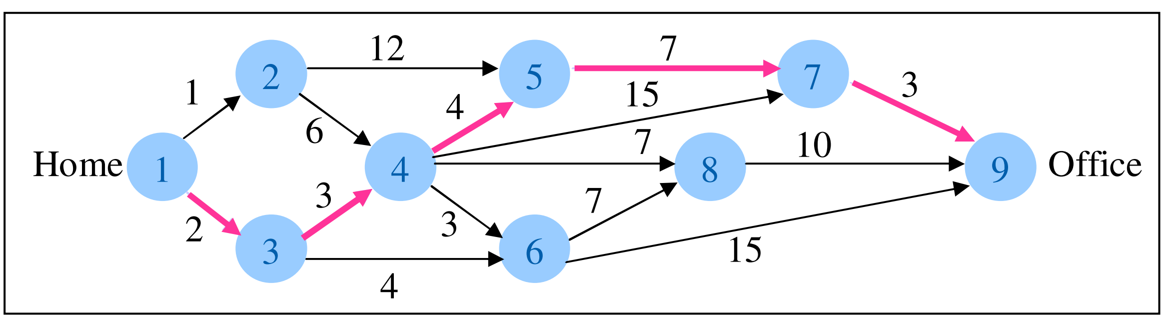

Figure 3.2 Optimal Movement Thru a Directed Network [24]

The problem states that John bicycles to work everyday. Node 1 represents his home and node 9 his office. The remaining nodes are road junctions. The number next to each arc is the travel time in minutes on the road it represents. All streets are one-way. The optimization problem is to find the quickest route from John's home to his office. Starting at node 9, we see that the path with the shortest travel time is arc (7, 9) with 3 minutes of travel time. Continuing backward, the fastest path to node 7 is the arc (5, 7) with 7 minutes of travel time. Then, the fastest path to node 5 is the arc (4, 5) with 4 minutes of travel time. The fastest path to node 4 is the arc (3, 4) with 3 minutes of travel time. The fastest (and only) path to node 3 is the arc (1, 3) with 2 minutes of travel time. Thus, the fastest path from home to office is (1, 3, 4, 5, 7, 9) with a total travel time of 19 minutes.

3.1.2 Dynamic Programming Example 2: A Multi-Stage Decision Process without Exogenous-Inputs

The following example, taken from [38], illustrates an *N*-step sequential-decision process in which the control decisions are a function of the current state. The state *x*, is a scalar with the following discrete state-evolution law

$$x_{k+1} = x_k + f_k D_k \,, \tag{3.1}$$

where k= 0, 1, …, (N-1), f_k is a known function different from zero and D_k is the decision variable at the current discrete time k. The criterion of optimality to be minimized is defined as

$$J = \left[x(T)\right]^2 + \sum_{i=0}^{N-1} D_i^2 \tag{3.2}$$

where T is the fixed terminal time at t_N. The first term in the optimization expression (3.2) indicates penalization of deviations of the state from the desired equilibrium while the second term discourages excessive use of the decision effort.

Using Dynamic Programming's backward recursive process in which the "starting-stage" is the *last* decision stage, (N-1), the solution for the optimal set of decisions $\left\{D_i^*\right\}_0^{N-1}$ begins by determining the "last" decision D_{N-1} first, at $k = (N-1)$ (the symbol * denotes an RTO-optimal decision). Thus, we solve for the optimal decision D_{N-1} at (N-1) for an *arbitrary* value of the process state x_{N-1} that gives the minimum J in the following one-step optimization

$$J_{N-1} = \min_{D_{N-1}}\left\{x(T)^2 + D_{N-1}^2\right\};\text{ for arbitrary } x_{N-1}. \tag{3.3}$$

According to the composite state evolution equation (3.1), $x(T = N)$ is related-to the current values being considered (x_{N-1}, D_{N-1}) by

$$x(T) = x_{N-1} + f_{N-1}D_{N-1}. \tag{3.4}$$

Substituting (3.4) into (3.3) gives J_{N-1} in terms of the variables $\{x, D\}$ at stage (N-1)

$$J_{N-1} = \left[x_{N-1} + f_{N-1}D_{N-1}\right]^2 + D_{N-1}^2. \tag{3.5}$$

From the continuity, smoothness and monotone-behavior of (3.5), it is clear that the absolute minimum of J_{N-1} with respect to the decision variable D_{N-1} exists [10] and corresponds to $dJ/dD_{N-1}=0$. Solving for the optimal decision $D_k{}^*$ at stage (N-1) gives

$$D_{N-1}^{*} = -\frac{f_{N-1}x_{N-1}}{1+f_{N-1}^{2}} . \tag{3.6}$$

Next, we move backward one additional time-step to the new "starting-time" $k=(N-2)$ and solve for the optimal decision at (N-2) for an *arbitrary* value of the state x_{N-2}. Thus, J_{N-2} becomes

$$J_{N-2} = \left[x(T)\right]^{2} + D_{N-1}^{2} + D_{N-2}^{2} \ . \tag{3.7}$$

Substituting $x(T)$ as given by (3.4) and using the previously computed D_{N-1} as given by (3.6) gives

$$J_{N-2} = \left[x_{N-1} + f_{N-1}\left(-\frac{f_{N-1}x_{N-1}}{1+f_{N-1}^{2}}\right)\right]^{2} + \left[-\frac{f_{N-1}x_{N-1}}{1+f_{N-1}^{2}}\right]^{2} + D_{N-2}^{2} . \tag{3.8}$$

According to the state evolution equation of (3.1), the value of the state x_{N-1} is determined by the known transition-equation

$$x_{N-1} = x_{N-2} + f_{N-2}D_{N-2} . \tag{3.9}$$

Thus, substituting (3.9) into equation (3.8) gives J_{N-2} in terms of the variables (x_{N-2}, D_{N-2}), as follows

$$J_{N-2}=\left[x_{N-2}+f_{N-2}D_{N-2}+f_{N-1}\left(-\frac{f_{N-1}\left(x_{N-2}+f_{N-2}D_{N-2}\right)}{1+f_{N-1}^{2}}\right)\right]^{2}$$
$$+\left[-\frac{f_{N-1}\left(x_{N-2}+f_{N-2}D_{N-2}\right)}{1+f_{N-1}^{2}}\right]^{2}+D_{N-2}^{2}. \tag{3.10}$$

Once again, since J_{N-2} is continuous, the absolute minimum of J_{N-2} with respect to the decision variable D_{N-2} exists and corresponds to the condition $dJ/dD_{N-2}=0$. We can solve for the optimal decision at stage (N-2) to obtain

$$D_{N-2}^{*}=-\frac{f_{N-2}x_{N-2}}{1+f_{N-2}^{2}+f_{N-1}^{2}}. \tag{3.11}$$

We continue this backward step-by-step process moving one additional backward time-step and solving for the corresponding D_k^* until we reach $k = 0$; in this way, we obtain a set of optimal decisions $D_k^*=D_k^*(x_k)$ for $k=(N-1),\ (N-2),\ \ldots,\ 1,\ 0$. Moreover, at each time k, the expression for D_k^* has the general form

$$D_k^{*}=-\frac{x_k f_k}{1+\sum_{k}^{N-1} f_k^{2}} \tag{3.12}$$

which allows calculation of the explicit expression for the optimal value of J in equation (3.2), in terms of x_0 alone.

3.1.3 Dynamic Programming Example 3: A Multi-Stage Decision Process with Exogenous-Inputs Modeled as Markov Processes

In this example, adapted from [2], the multi-stage decision problem has a state evolution equation that is being influenced by an uncertain, uncontrollable exogenous-input w_k acting on the system. The corresponding state evolution equation is

$$x_{k+1} = ax_k + bD_k + w_k\,, \tag{3.13}$$

where k= 0, 1, …, (N-1), a and b are known constants different from zero and D_k is the decision variable acting on the system at the current discrete time k. The terminal time T is fixed at $k = t_N$. Because w_k can affect the x_i-values, it is important to use as much information about w_k as is "known" or "knowable" at each k= 0, 1, …, (N-1). Exogenous-inputs in sequential-decision processes are traditionally modeled as *random disturbances* with known, assumed reliable, probabilities (such as long-term mean and variance). In the previous example there were no disturbance-inputs and thus, no uncertainty. In that case, when a decision is decided for a given state x_k, the next state x_{k+1} is fully and precisely determined. This is known as a *deterministic* problem [16]. In this example, because of the presence of the uncertain, unmeasurable exogenous-input w_k and the way it has traditionally been modeled as a random disturbance characterized by a probability distribution, the problem becomes non-deterministic. In this example, w_k is considered a random parameter and in turn the process state x and the criterion of optimality J become random variables that cannot be meaningfully optimized in the strict-sense [16]. The random exogenous-input w_k of this example, has traditionally been

characterized by a probability distribution $P_k(\cdot \mid x_k, D_k)$ [11], [22], [25], [27] which may depend explicitly on the current value of the process-state, x_k, and the current value of the decision variable, D_k, but not on past values of the exogenous-input $w_0,\ w_1,\ \ldots, w_{k-1}$ (first-order Markov-modeling assumption) [1], [26]. Thus, the performance-functional (3.2) can only be "optimized" in some random-variable context, such as "expected-value" over a large ensemble of "sample-functions." Therefore, in that case, the Dynamic Programming solution of optimal "sequential-decision" problems in the face of uncertain first-order Markov-type exogenous-inputs reduces to the optimization of the *expected-value* of the performance-functional, i.e.,

$$J = E\left\{ g_N + \sum_{k=0}^{N-1} g_k(x_k, D_k, w_k) \right\};\ E = \text{expectation operator,} \tag{3.14}$$

where the expectation is with respect to the (presumed-known) joint-distributions of all the random variables involved [16], [24].

Continuing with the present example, suppose the exogenous-input values w_k are defined as a set of random variables that are independent and have the following known probabilities in the form of large-ensemble, long-term mean and variance

$$\begin{aligned} &E\{w_k\} = 0 \\ &E\{w_k^2\} = \sigma_k^2,\ 0 \le k \le N-1, \end{aligned} \tag{3.15}$$

where σ_k^2 denotes the value of the known variance of w_k.

Applying the Dynamic Programming backward recursive process as in Example #2 and starting at the last decision stage, (N-1), we can solve for the optimal decision at (N-1) for an *arbitrary* value of the process state x_{N-1} that minimizes the expected value of J, which for this example has the following form

$$J_{N-1} = \min_{D_{N-1}} E\left\{x(T)^2\right\} \ . \tag{3.16}$$

According to the composite state evolution equation (3.13), $x(T = N)$ is related-to the "previous values" (x_{N-1}, D_{N-1}) by

$$x(T) = x_N = ax_{N-1} + bD_{N-1} + w_{N-1} \, . \tag{3.17}$$

Substituting (3.17) into (3.16) gives J_{N-1} in terms of the variables {x, D} at stage (N-1)

$$J_{N-1} = E\left\{\left[ax_{N-1} + bD_{N-1} + w_{N-1}\right]^2\right\}. \tag{3.18}$$

As mentioned in (3.14), the notation E{.} corresponds to the expectation operation. In [2], Aoki presents a useful formula using the expectation operation: given a random variable x, we can write

$$E\left\{x^2\right\} = \left[E(x)\right]^2 + \mathrm{var}(x)\,, \tag{3.19}$$

where var(x) is the variance of x. After applying (3.19) to (3.18) J_{N-1} becomes

$$J_{N-1} = \left[E\left\{ax_{N-1} + bD_{N-1} + w_{N-1}\right\}\right]^2 + \mathrm{var}(x_N)\,, \tag{3.20}$$

where x_N is the value of the process state at the terminal time *T*=*N*. Using the mean and variance from (3.15), $E\{x_N\} = a \cdot x_{N-1} + b \cdot D_{N-1}$ and $\text{var}(x_N) = \sigma_{N-1}^2$ and J_{N-1} becomes

$$J_{N-1} = \left(ax_{N-1} + bD_{N-1}\right)^2 + \sigma_{N-1}^2. \tag{3.21}$$

The absolute minimum of J_{N-1} with respect to the decision variable D_{N-1} exists and J_{N-1} is minimized with respect to the decision D_{N-1} which gives

$$\frac{dJ_{N-1}}{dD_{N-1}} = 2\left(ax_{N-1} + bD_{N-1}\right)b = 0, \tag{3.22}$$

since σ_{N-1} is independent of D_{N-1}. Solving for the optimal decision at stage (*N*-1) gives

$$D_{N-1}^{*} = -\frac{ax_{N-1}}{b}. \tag{3.24}$$

The decision policy in (3.24) for this example turns out to be deterministic (no uncertainty) while the *E[J]* in (3.21) is **increased** by an amount proportional to the variance of the exogenous-input [2]. This process can be repeated for the other stages $(N-2), \ldots 1, 0$ to obtain a set of optimal decisions at each stage in terms of the process state x_k at each stage *k*.

3.2 The Traditional Markov-Modeling Procedure for Uncertain Exogenous-Inputs

The modeling of exogenous-inputs has developed through history in different areas. In the area of control systems, Classical Control, which spans from 1938 to 1958, produced integral controllers, feedforward controllers, and notch filters capable of coping

with exogenous-inputs such as steps, ramps, sinusoids, etc., in scalar time-invariant systems. From 1958 to present day, Modern Control has introduced techniques that use state variable approach and optimal control to study multivariable control problems. However, these techniques have been slow in coming up with ways to cope with disturbances (exogenous-inputs). In 1968, Johnson introduced [30] the Disturbance Accommodating Control (DAC) theory [31]-[33] which is a multivariable feedback control technique that generalizes "integral," match-filter" and other classical techniques for coping with non-noisy type disturbances encountered in practical applications. The DAC technique uses the state-variable format and introduces the concept of a disturbance-observer. The non-noisy types of disturbances that DAC-Theory is designed-for are called waveform or structured-variation (S.V.) type disturbances [35].

Parallel to the development of control techniques to cope with exogenous-inputs, statistical approaches have been used to represent and develop means for mitigating the effects of disturbances. Andrey Markov produced the first theoretical results in 1906 [3]. His results were later generalized by Kolmogorov (1936) [3] and they were followed by Brownian motion [19], a topic which was important in the early years of the twentieth century and became a popular method for modeling stock market fluctuations [46]. The modern study of stochastic sequential decision problems began with Wald, around the Second World War, who first published his results in 1947. Branching out from operations research in the 1950s, Markov decision process models emerged. They quickly gained recognition in diverse fields such as ecology, economics, and communications engineering. All of those models represented exogenous-inputs by means of noise-type models [48], a widely accepted practice which continues today and

has been traditionally employed in sequential-decision problems involving uncertainty [13], [14], [16], [24].

The classical technique for representing the effects of uncertain exogenous-inputs (disturbances) w_k in sequential decision problems is to model them as discrete-time Markov processes with reliably-known parameter-values as shown in Example #3 of the previous section. Markov processes are characterized qualitatively by their noise-type and erratic nature (see Figure 3.3). Thermal noise in radar and radio receivers is an example of this noise-type exogenous-input. In particular, the time-behavior of the exogenous-input w_k is defined in terms of its *long-term* statistical properties, such as mean-value, covariance, power spectral density, etc., which are assumed to be (somehow) reliably-known *a priori*. In this way, noise-type exogenous-inputs are mathematically modeled by traditional random-process theories which, as is well-known [16], [24], [28], yield effective models, if one can assume that the exogenous-inputs have time-variations that can be viewed as *sample-functions* belonging to an extensive *ensemble* of "similar" functions that obey (and are thus limited by) the "Ergodic-hypothesis" [28], [47]. The Markov models used to represent such noise-type exogenous-inputs w_k are typically expressed as first-order or second-order "coloring filters" with "white noise" inputs. In discrete-time applications, the first-order versions of those coloring-filters are typically assumed to have the following *difference-equation* format [16]:

$$w_{k+1} = \lambda \cdot w_k + \xi_k \;\; ; \; \xi_k = \text{discrete-time white noise}, \qquad (3.25)$$

where k is the (real-time) discrete-time index, w_{k+1} is the output of the (discrete-time) Markov coloring-filter at the "next" time (k+1) and represents the (expected) real-time

value of the uncertain exogenous-input acting-on the underlying process, λ is a given scalar selected to produce certain long-term statistical/probability properties of $\{w_k\}$ such as power spectral density (psd), and the discrete-time "white-noise" ξ_k is a time-sequence of independent random numbers with a specified distribution of arrival-times and corresponding values, both assumed reliably-known *a-priori*.

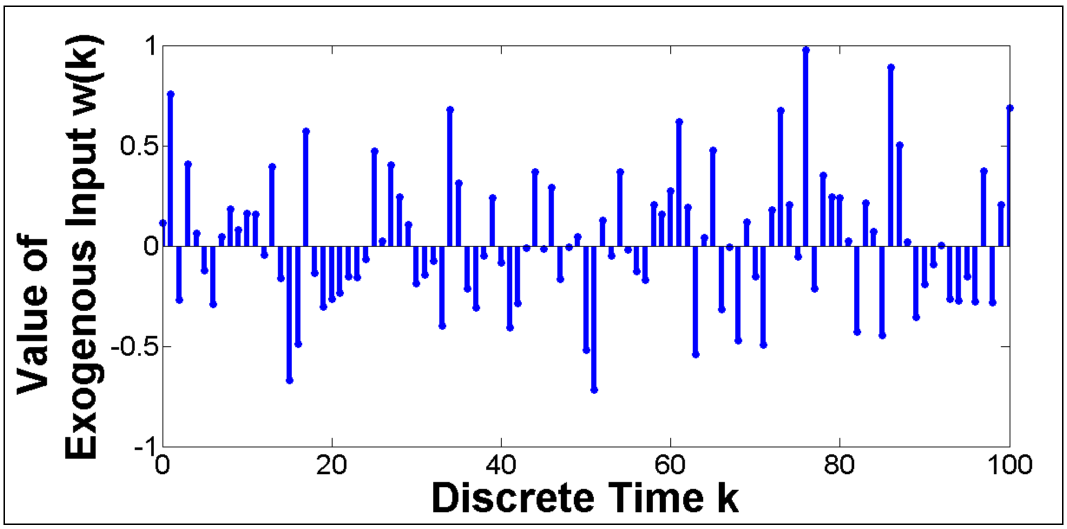

Figure 3.3. Typical Plot of a Noise-Type Discrete-Time Exogenous-Input

3.3 Waveform-Structured Time-Variations of Uncertain Exogenous-Inputs and the Structured-Variation Technique for Modeling Uncertain Exogenous-Inputs

In a series of papers beginning in 1968 [30]-[33], the concept of modeling uncertain exogenous-inputs having "structured" time-variations (aka waveform-structured variations) was introduced. That new concept represented a sharp departure from the traditional Markov models and was intended for modeling the class of

exogenous-inputs w_k that are unknown in the quantitative-sense but are known to have time-variations that behave in certain qualitative patterns (characteristic patterns of time-behavior). For example in some cases, the exogenous-input (essentially) behaves as an unknown "constant" or "bias" that "changes"-value only "occasionally" (=sparse-in-time), is typically non-Ergodic, and thus cannot be accurately represented by a conventional noise-type Markov model. In these cases, "smarter"/"wiser" decisions in a sequential-decision process can be made if information about the *short-term* or current value of the "constant" exogenous-input is known, which *long-term average* statistical properties do not reveal, with any confidence. As already mentioned, these types of uncertain exogenous-inputs have been extensively studied in [30]-[33] and today are known as inputs with *structured-variations* or *waveform-structure* [5]. The waveform modeling technique developed in [35] and [30]-[33], enables one to describe, a priori, the *range* of possible time-variation patterns a particular unknown exogenous-input can exhibit at any moment of time (see Figure 3.4). Those patterns of time-variations can be modeled using an analytical expression known as a *spline-function* which, in the linear, discrete-time case has the form

$$w(k) = c_1(k) \cdot l_1(k) + c_2(k) \cdot l_2(k) + \ldots + c_M(k) \cdot l_M(k)\,, \tag{3.26}$$

where k is the discrete time index, the $l_i(k)$, $i=1, \ldots, M$ are known functions (=modes of behavior) and the $c_j(k)$, $j=1,\ldots, M$ are *unknown*, "stepwise constant" weighting-coefficients that, at each time k, blend the known modes in a weighted, linear-combination sense to create the "current" value and local behavior of $w(k)$. The values of the $c_j(k)$ are assumed to only change at *sparse*, unknown times [35].

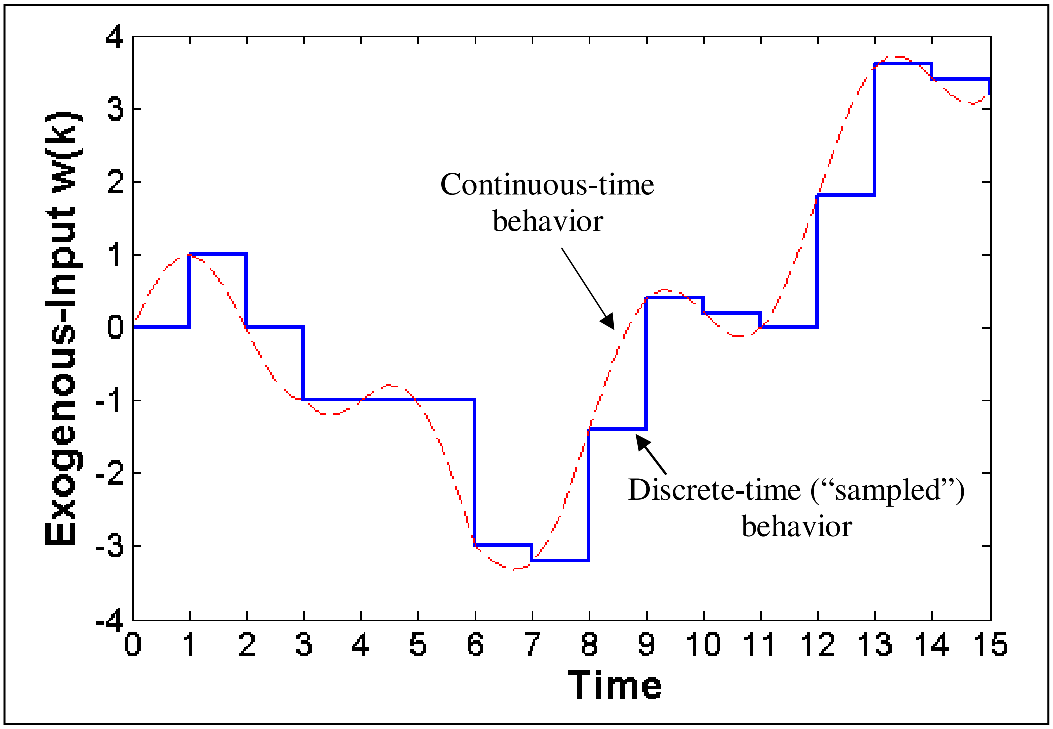

Figure 3.4. Example of the Conventional Representation of Continuous vs. Discrete-Time Behavior of a Structured-Variation Exogenous-Input *w(k)*

The values of waveform-structured exogenous-inputs mathematically modeled by the linear analytical expression of (3.26), are composed of weighted-sums of a collection of known/chosen functions $l_1,\ldots,l_M$ which play the role of *basis functions* and a collection of unknown, "stepwise-constant" weighting-coefficients $c_1,\ldots,c_M$ that change values in a time-sparse manner. Thus, at any moment k, an unknown, structured-variation type exogenous-input is expressed as a linear combination of the *known* basis functions $l_i(k)$ having unknown weighting-coefficients $c_j(k)$. The $l_i(k)$ are determined by the underlying process that "generates" the exogenous-input $w(k)$ and can be identified by testing, experimental data, and known "laws"/truisms of physics, economics, etc.

In structured-variation modeling, the traditional statistical properties and assumptions that are essential to the success of Markov-models can be disregarded altogether, since they are irrelevant to the representation of the characteristic time-behavior of uncertain exogenous-inputs that have known "waveform structure." The waveform-structure characterization is applicable to Ergodic as well as highly *non-Ergodic* uncertain exogenous-inputs and allows the design of a physically realizable deterministic-type sequential-decision algorithm and associated "state-observer" which can be remarkably effective in optimizing performance *in a non-gambling sense*, when faced with uncertain structured-variation type exogenous-inputs.

In the rather common case of "linear-dynamic" (L. D.) type [38] basis-functions in (3.26), the dynamic state z of the exogenous-input $w(k)$ in (3.26) can be chosen such that in the general multi-variable "exogenous-input" case $\{ w_1, w_2, \ldots, w_M \}$, <u>each</u> exogenous-input w_j is assigned an "exostate-vector" $z_j(k)$ to obtain the following linear multivariable-input discrete-time "exostate" model [35]

$$\begin{aligned} w_j(k) &= H(k) z_j(k); \\ z_j(k+1) &= G(k) z_j(k) + \overline{\sigma}(k), \end{aligned} \tag{3.27}$$

where z_j is a ρ_j-dimensional vector representing the "state" ("exostate") of the structured-variation exogenous-input "component" $w_j(k)$, H and G are known $1 \mathrm{x} \rho_j$ and $\rho_j \mathrm{x} \rho_j$ matrices, respectively, selected to produce the known characteristic waveform features of $w(k)$ (also written as w_k), and $\overline{\sigma}$ is a ρ_j-vector of *time-sparse* sequences of Kronecker delta functions with completely unknown arrival times and values which cause the

corresponding, unknown, sparse-in-time jumps in the values of the $c_j(k)$ in (3.26), via corresponding jumps in the individual elements of z_j [34].

The structured-variation model (3.27) has the same outward appearance as the Markov model in (3.25). However, they differ in several important ways. Namely, the term ξ_k which is "white noise" in Markov modeling is replaced by a sequence of *time-sparse* Kronecker delta functions $\bar{\sigma}$ in structured-variation modeling. They also differ in that the quantity λ in (3.25) is selected to produce certain long-term averaged power spectral densities in Markov modeling, whereas in (3.27), the known matrices H and G are selected to produce certain characteristic <u>time-variation</u> features of the "sample-functions" modeled by (3.26) and the associated structured-variation "state-model" (3.27).

We emphasize that the sequence $\bar{\sigma}$ **cannot** be accurately described by common statistical characteristics since it is not an Ergodic random process in the usual sense, because the jumps in $\bar{\sigma}$ are "sparse-in-time." As an example of this type of non-noisy exogenous-input w_k, let w_k be a "stepwise-constant" and imagine a sequential-decision process with ten decision stages and for which we know that there will exist at most *only* one Kronecker delta function $\bar{\sigma} \Rightarrow w_k$=constant. We simply know that there will be (at most) *one* "jump" in the value of w_k but we do not know <u>when</u> it will occur or its <u>value</u>. This type of uncertain, unknown exogenous-input cannot be accurately modeled using ordinary statistical metrics. However, w_k has "structured-variations" with the one simple basis-function $l_1(k)$=1 and it can be effectively modeled and "observed" using the state modeling technique described in this dissertation and in [20], [30]-[33], [35].

Note that if it should turn-out that with structured-variations (S. V.-type) $\overline{\sigma} \equiv 0$ in (3.27), the sets of coefficients $\{c_k\}$ in (3.26) remain constant and in retrospect there was not any uncertainty associated with the future time-behavior of the exogenous-input w_k. Thus, if $\overline{\sigma} \equiv 0$, the future time-behavior of w_k is knowable at each future time k', $k \le k' \le T$, where T is the time at the end of the sequential-decision process. On the other hand, if $\overline{\sigma}_k \neq 0$ in (3.27), the value of $z(k)$ (also written as z_k) can still be known to the decision maker in real time but this does not imply knowledge of z_k at any future time k', $k \le k' \le T$.

In "structured-variation" modeling and related DAC control-algorithm design, real-time "composite-state" observers [35; p. 241] are used to determine the real-time values of the "state" z_k of the actual exogenous-input w_k at *each* decision time. It should be noted that the state z_k of an exogenous-input is an abstract "information-quantity" comparable to the "state" of a dynamic system. Nevertheless, an optimal decision based on the instantaneous value z_k of the state of the exogenous-input w_k along with the value of the process state x_k embodies all the information needed to make a rational, real-time decision at time k (called a "real-time optimal" decision), even though the future behavior of the exogenous-input is unknown for all times greater than k [38].

Figure 3.5 is an illustration of the time-variations of a structured-variation type exogenous-input having the characteristic pattern of time-behavior which consists of a linear combination of "steps and ramps" as represented by

$$w_k = w(k) = c_1(k) \cdot 1 + c_2(k) \cdot k, \tag{3.28}$$

where $c_1(k)$ and $c_2(k)$ are unknown stepwise-constant weighting coefficients whose values vary in a random fashion with jumps that are *sparse-in-time* as shown in Figure 3.6. The known basis functions of the structured-variation exogenous-input of (3.28) are

$$l_1(k) = 1, \quad l_2(k) = k \ . \tag{3.29}$$

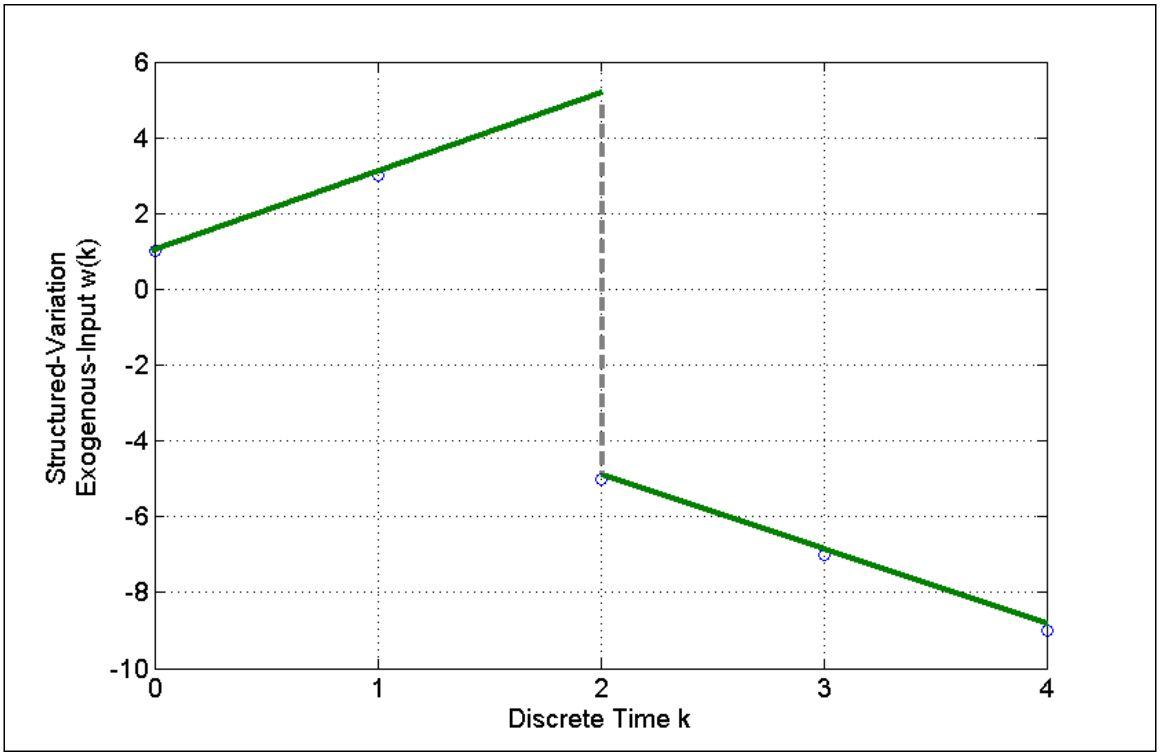

Figure 3.5. Example of the Time-Behavior of the Waveform-Structured Exogenous-Input of (3.28)

To develop an effective state-model for a structured-variation exogenous-input w_k modeled by the spline expression (3.28), one must first find the minimal-order difference equation for the exogenous-input spline-model. Then, one can seek a proper state vector for that difference equation which leads to the sought state-evolution model. The minimal-order, homogenous difference equation for which (3.28) is the general solution (see Appendix D for Step plus Ramp type exogenous-input) is found to be

$$w(k+2) = 2w(k+1) - w(k)\,. \tag{3.30}$$

The difference equation (3.30) leads-to the following discrete state-space model of the exogenous-input with structured-variation (3.28)

$$\begin{aligned} w(k) &= \overbrace{\begin{bmatrix} 1 & 0 \end{bmatrix}}^{H \text{ in } (3.27)} \begin{pmatrix} z_1(k) \\ z_2(k) \end{pmatrix}; \qquad \begin{aligned} z_1(k) &\underline{\underline{\Delta}} w(k) \\ z_2(k) &\underline{\underline{\Delta}} w(k+1) \end{aligned} \\ \begin{bmatrix} z_1(k+1) \\ z_2(k+1) \end{bmatrix} &= \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}}_{G \text{ in } (3.27)} \begin{pmatrix} z_1(k) \\ z_2(k) \end{pmatrix} + \begin{pmatrix} \sigma_1(k) \\ \sigma_2(k) \end{pmatrix}, \end{aligned} \tag{3.31}$$

where $\sigma(k) = \sigma_k$, $w(k) = w_k$ and $z(k) = z_k$. It should be noted that the <u>dimension of the exostate</u> "z" will always be $\geq$ the number M of basis-functions in (3.26).

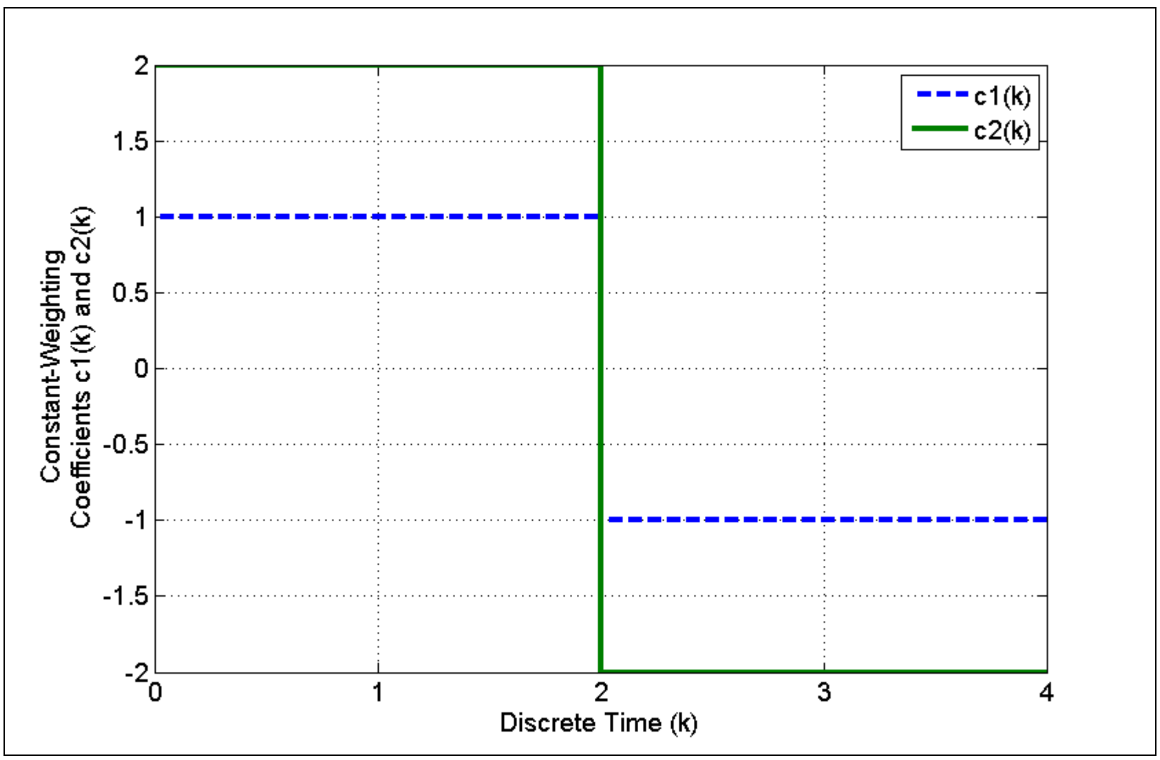

Figure 3.6. Time-Behavior of Stepwise-Constant Weighting Coefficients $c_j(k)$ of (3.28)

3.4 The New Principle of "Real-Time Optimality" (RTO)

In the 2005 paper titled "Beyond Bellman's Principle of Optimality: The Principle of Real-Time Optimality (RTO)" [38], it was shown that Bellman's Principle of Optimality may not apply to Dynamic Programming techniques involving uncertain exogenous-inputs with "structured-variations." According to the Principle of Optimality, the information embodied in the current value of the problem-state x (this "state," vaguely-defined in most of the "dynamic programming" literature, is called the "Bellman state" in [38]), at any time t (or discrete time k), must be sufficiently rich in quantitative information about "future consequences of current decisions" to enable an *absolutely optimal* decision. However, when the uncertain exogenous-input has waveform structure (S. V.-type), the Dynamic Programming solution process *violates* the constraint that the *future* time-behavior of the real-time values of the class of exogenous-inputs w_k with structured-variations that we are considering are not known (or even "knowable") *a priori* and thus are not reliably predictable [38]!

In [38], it was shown that if the uncertain variation of an exogenous-input w_k can be modeled by the spline-function (3.26), one can introduce an exostate state "z" and corresponding "exostate" state-model for the exogenous-input w_k and its equivalent state model (3.27). The value of the exostate z, together with the underlying process state x form a composite (augmented) state

$$\tilde{x}_k = col.(x_k \mid z_k); \quad col.\text{=column vector} \tag{3.32}$$

that turns-out to behave as a "Bellman-state" *in the RTO sense*, described below. When the exogenous-inputs have "structured-variations", and the composite state $\tilde{x}_k$ is completely-observable in the sense of Kalman [18], real-time "composite-state" observers [35; p. 241] can be used to determine the real-time values of the "state" z_k of the actual exogenous-input w_k at *each* decision time. The composite state of (3.32) constitutes a "valid" state which enables a rational determination of "Real Time Optimal" decisions at each time k. This is known as the Principle of "Real Time Optimality (RTO)" [38]. RTO decisions use the maximum available information, at each decision-time, while the underlying dynamic process is unfolding and without any knowledge of (or gambling-about) unforeseen, time-sparse "events," that may, or may not, occur in the future. In the sense of RTO, the composite state $\tilde{x}$ contains all the information necessary to describe the future plant-state evolution and the future behavior of w_k, <u>assuming</u> the $\{\bar{\sigma}_k\}$ in (3.27) are zero, from that moment on. At any stage k, when the "sparse-in-time" $\bar{\sigma}_k$ arrive, the instantaneous value of z_k and the instantaneous value of x_k together embody all the information needed to make a rational, *real-time* "RTO decision" which is the best attainable in the presence of the unpredictable and unexpected time-sparse "surprises" $\bar{\sigma}_k$. The RTO decision does not attempt to minimize the expected value of the cost functional (3.14) (recall the σ_k are non-Ergodic, in general), but instead minimizes the cost functional of (2.2) at every single stage k assuming the current trend in w_k -behavior remains unchanged ($\bar{\sigma}_k \equiv 0$) <u>and</u> updating that latter assumption at <u>each</u> successive decision-time k, using the z_k-value that exists <u>at that new decision-time</u>.

As previously mentioned, knowledge of the value of z_k at the current time k is equivalent to knowledge of the current values of the set of constant coefficients $\{c_k\}$ in

(3.26). However, because of the possible future arrivals of the time-sparse $\bar{\sigma}_k$ in (3.27), knowledge of the current value of z_k does not imply knowledge of the future values of the exogenous-input $w_{k'}$ at the future times *k'* ($k \le k' \le T$). On the other hand, if the values of the $\{c_k\}$ remain identically-constant at the "current" value (equivalent to $\bar{\sigma}_k \equiv 0$ for all future times *k'*, $k \le k' \le T$), then knowledge of the value of the composite state $\tilde{x}_k = col.(x_k \mid z_k)$ would enable determination of the absolute optimal decision at each time *k*. In the case of non-noisy, uncertain, structured-variation exogenous-inputs, RTO-optimal decisions are based on the "current" or instantaneous values of x_k and z_k at each decision time *k* tacitly assuming $\bar{\sigma}_k \equiv 0$ in (3.27) for each future time *k',* $k \le k' \le T$ and updating that assumption (and compensating for any Kronecker delta functions $\bar{\sigma}$ that have arrived since the last RTO decision-time). In this way, the composite state of (3.32) constitutes a "valid" state for the Dynamic Programming method and enables a rational, non-gambling, determination of "Real Time Optimal" decisions at each time *k*.

Since, in reality, $\bar{\sigma}_k$ rarely equals zero for all current and future times and the *time–sparse* arrival times, number of and intensities for $\bar{\sigma}_k$ are totally unknown and unpredictable, the degree of optimality of the RTO decision D_k is, typically, *not* strictly optimal in the absolute sense. However, the optimality of an RTO-optimal decision can *only* be improved by lucky guesses or "fortuitous gambling" about the future arrival times and values of the sequence of (a finite but unknown-number of) *time-sparse* Kronecker delta-functions denoted by $\bar{\sigma}_k$ in (3.27). In practical applications, such a "gamble" about a finite-number of isolated events, that may or may not occur, often involves very undesirable consequences, regarding the effects-on the value of *J*, if the

gamble is not successful (not fortuitous). For that reason, RTO-decisions seem to be the natural default decisions seen in most “competitions” involving humans and in virtually all “competitions” involving predator-pursuit/victim-evasion “strategies” that naturally occur in minor (lower) forms of life (animals, birds, reptiles, fish, insects, etc.) as explained in [38].

RTO-type decisions, when “structured-variation” exogenous-inputs are acting on the underlying process, can be derived using a modification of the Dynamic Programming methodology. This is accomplished by applying the concepts of Waveform Modeling to represent the qualitative features of the structured-variation type exogenous-input, the exostate z_k to represent the state of the exogenous-input and by introducing a composite state $\tilde{x}_k = col.(x_k \mid z_k)$ which, in terms of the RTO Principle, provides a valid state for a Dynamic Programming-type solution process. The next chapter will illustrate how one can systematically incorporate “structured-variation” type exogenous-inputs into a Dynamic Programming-type solution process by invoking the more-general RTO Principle in place of Bellman’s Principle of Optimality, which will be seen to be a *special-case* of RTO, where the $\sigma_k \equiv 0,\ \forall k > 0$.

Chapter 4

A DYNAMIC-PROGRAMMING TYPE SOLUTION PROCEDURE FOR RTO-TYPE OPTIMAL DECISIONS IN SEQUENTIAL-DECISION PROCESSES WITH STRUCTURED-VARIATION EXOGENOUS-INPUTS

In the previous chapter, we described a particular class of uncertain exogenous-inputs that do not satisfy the assumptions underlying the Markov-modeling procedure traditionally employed in SDP. That class of uncertain exogenous-inputs is known as "exogenous-inputs with structured-variations" [35]. We showed that there is an alternative modeling procedure for that particular class of exogenous-inputs which has the benefit of enabling the modeling, and real-time estimation, of such exogenous-inputs by state-variable modeling and modern Kalman State-Observer/Filtering [42] and estimation techniques assuming the system state x_k and the "state" z_k of the structured-variation input w_k are <u>completely</u> <u>observable</u> in (real) discrete-time. Using that technique and invoking the new [38] and more general Principle of Real-Time Optimality (RTO), it is shown in this chapter that, after introducing a "new" augmented state-vector, one can use Bellman's backward-solution procedure to derive sequential-decisions that are "optimal" in the more-general RTO sense. Because the "states" z_k of w_k, and x_k, are both

updated to the correct values at each decision-time, the RTO Principle yields the "best" rational (non-gambling) decision strategy one can follow when the exogenous-inputs have structure-variations, as we have described here.

4.1 A First-Order Example of a Finite SDP

Using the augmented-state notation of (3.32), it was shown in [20] how uncertain structured-variation type (LD) exogenous-inputs can be modeled by a linear state-variable model that can be incorporated into the traditional Dynamic Programming solution technique to obtain a set of sequential-decisions that are optimal in the "RTO sense," which is more-general than Bellman's Principle of Optimality. This procedure is best illustrated by considering an exceptionally simple, specific SDP example in which the process at each stage is described by the *same* dynamical model. Thus, we consider a simple process modeled by a first order, discrete-time state-evolution equation which describes how both the "current" state of the underlying dynamic process (and the effect of the "current" exogenous-input w_k) alter the sequential transition of the process state x from one stage (k) to the next stage (k+1). That process state-model is assumed to have the form of the general, linear first-order difference-equation

$$x_{k+1} = a_k x_k + b_k D_k + f_k w_k \,, \tag{4.1}$$

where the sequential "stages" (indexed by k) in the sequential decision problem extend over the finite time interval $t_0 \le k \le T$, $T = fixed < \infty$, x_k is the scalar process-state, D_k is the scalar decision determined at each decision-time k, w_k is an unknown, unmeasurable, and uncontrollable scalar exogenous-input, and the coefficients (a, b, and f) are also scalar and known, arbitrary real functions, which, in principle, can change with time in

some manner (assumed known a priori). To simplify the calculations, we will consider the *time-invariant* case (a, b, and f = known constants). As mentioned in the previous chapter, according to the RTO Principle, the "optimal" decisions D_k are based on the instantaneous-values of the composite-state $\tilde{x} \underline{\underline{\Delta}} (x, z)$ at each time t_k, where, according-to the RTO Principle, one presumes, at each decision-time, that the values of the $c_j(k)$ *will remain constant* in (3.26) for all future $t_k \geq k$, or equivalently that $\bar{\sigma}_k \equiv 0$ in (3.27) for all future times $t_k \geq k$. It is critically-important to recall that this assumption is a "tacit-assumption" and is updated at each "decision-time" t_k and thereby "RTO optimally" partially "corrects" for any changes in the "constant" $c_j(k)$-values that have occurred since the last decision-time. This tacit-assumption, and its recursive-correction at each new decision-time, enables a set of sequential, rational decisions that do not involve "gambling" about time-sparse events (future jumps in the $c_j(k)$-values which are <u>inherently non-Ergodic</u> and thus cannot be reliably-characterized by examination-of one or more "sample-functions " of w_k-behavior). Thus, as already mentioned, knowledge of the $\tilde{x}$-values at each decision time t_k provides sufficient information to enable a set of rational, *real-time decisions* that are "optimal" in the RTO sense [38] and cannot be improved except by fortuitous-outcome of a "gambling-process" regarding time-sparse, non-Ergodic events.

Suppose, in this example, the scalar-valued criterion of optimality J to be *minimized* for the sequential decision process (4.1) is the sum of the square of the final state $x(T)$ of the process and the square of the individual decisions D_k, $k=t_o,\ldots,(N\text{-}1)$ and has the following specific form

$$J = x_T{}^2 + \sum_{k=t_o}^{N-1} (D_k^2) \ , \tag{4.2}$$

where the first term on the right-side of the optimization expression (4.2) is placed there to penalize deviations of the terminal-state value x_T from the desired value $x_T = 0$ (the state x_T in (4.1) often corresponds to the "terminal-miss" of some economic or other "target value" the decisions D_k are striving to achieve) while the second term on the right-side of (4.2) discourages excessive magnitude/intensity of the decisions D_k.

Suppose further that the characteristic nature of the uncertain time-behavior of the scalar exogenous-input w_k, has been studied and modeled by the structured-variation technique, leading to the following first-order, discrete-time, exostate-model

$$\begin{aligned} w_k &= h z_k; \ \ (h = 1.0) \\ z_{k+1} &= g z_k + \overline{\sigma}_k \ , \end{aligned} \tag{4.3}$$

where the coefficient g denotes an arbitrary, real, known constant. For example, by setting $g=1$ in (4.3), the case of an unknown "stepwise-constant"-type exogenous-input is obtained, i.e., $w_k = C$, where C represents an unknown "constant" that may abruptly change value in an unknown time-sparse manner.

The RTO concept as applied in [20] for solving the optimal sequential-decision problem of (4.1)-(4.3) for arbitrary initial conditions { x_{to} , z_{to}}, can be implemented by first defining the *composite state* $\tilde{x}$ consisting of the underlying system (process) state x and the structured-variation type exogenous-input "exostate" z, and then developing the

discrete time state-evolution model for the composite state $\tilde{x} \underline{\underline{\Delta}}\, col.(x|z)$. The latter result is

$$\tilde{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} a & fh \\ 0 & g \end{bmatrix} \cdot \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} \cdot D_k + \begin{bmatrix} 0 \\ \sigma_k \end{bmatrix}; \;\; \tilde{x} \underline{\underline{\Delta}} \left(\frac{x}{z} \right), \tag{4.4}$$

where x is the scalar state process, $k= t_o,\ldots, (N\text{-}1)$, $N=T$, and z_k is the "current" value of the "exostate" z of the structured-variation type exogenous-input w_k, as modeled by (4.3). According to the RTO Principle [38], at each decision time k, the "RTO-optimal" decisions D_k are based on the instantaneous-values of the augmented state $\tilde{x}$, as though one were **tacitly-presuming** that all the $c_j(k)$ remain *constant* in (3.26), or equivalently that $\bar{\sigma}_k \equiv 0$ in (4.3) and (4.4), *for all future times* $t_k \geq k$. As previously mentioned, this tacit-assumption is updated (and thereby unpredictable changes in the c_j -values are compensated-for, to the maximum extent possible, given the sparse-nature of the c_j -changes) at each successive decision-time k, and in this manner the value of $\tilde{x} = col.(x \mid z)$ provides sufficient information at each decision-time to enable a set of rational, *real-time decisions* $\{D_k\}$ that are "optimal" in the RTO sense [38] and can only be improved (at the real-time moment) by a fortuitous-guess/gamble about the number-of, the times of occurrence and individual "values" of a sequence of time–sparse future events that may or may not occur (and thus are inherently non-Ergodic, in general)!

As explained in Chapter 3, in the backward-recursive Dynamic Programming technique, the "starting-stage" is the *last* decision stage, t_k=k=(N-1). Thus, following that protocol, we solve for the RTO decision at t_k=k=(N-1) that gives the minimum value of J for an *arbitrary* value of the process state x_{N-1} and an *arbitrary* value of the "exostate"

z_{N-1}. Thus, the solution for the RTO-optimal sequence of decisions, D_k* with k=(N-1), (N-2), …, t_o (the symbol * denotes an RTO-optimal decision) begins at k=(N-1) with the *one-step* optimization of the performance index (4.2) written in the following equivalent form

$$\tilde{J}_{N-1} = \min_{D_{N-1}} \left\{ \tilde{x}_T^T S \tilde{x}_T + D_{N-1}^2 \right\} \; ; S = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \tag{4.5}$$

where S is a given (user chosen) non-negative definite matrix of dimension $(n+\rho)\times(n+\rho)$, and $\tilde{x}^T$ denotes the transpose of the composite-state $(x|z)$.

According to the composite-state evolution equation (4.4) and the RTO tacit assumption $\overline{\sigma}_k = 0$ for all "future" k, $\tilde{x}_T$ is related-to the previous values ($x_{N-1}, D_{N-1}, z_{N-1}$) by the completely deterministic relation

$$\tilde{x}_T = \tilde{x}_N = \begin{bmatrix} a & fh \\ 0 & g \end{bmatrix} \tilde{x}_{N-1} + \begin{bmatrix} b \\ 0 \end{bmatrix} D_{N-1} \; ; \; \tilde{\sigma}_{N-1} = 0, \;\; \tilde{x} \underline{\underline{\Delta}} \left(\frac{x}{z} \right), \tag{4.6}$$

Substituting (4.6) into (4.5) and multiplying the constant coefficient matrix and vector $\tilde{x}_{N-1}$ gives $\tilde{J}_{N-1}$ in terms of the variables $\{x, D, z\}$ at stage (N-1)

$$\begin{aligned} \tilde{J}_{N-1} &= \min_{D_{N-1}} \left\{ \begin{pmatrix} x_T, & z_T \end{pmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{pmatrix} x_T \\ z_T \end{pmatrix} + D_{N-1}^2 \right\} \\ &= \min_{D_{N-1}} \left\{ \left(a x_{N-1} + fh z_{N-1} + b D_{N-1} \right)^2 + D_{N-1}^2 \right\}. \end{aligned} \tag{4.7}$$

From the structure of (4.7), it is clear that the absolute minimum of the composite $\tilde{J}_{N-1}$ with respect to the decision variable D_{N-1} is well defined, exists, and is determined

by $d\tilde{J}_{N-1}/dD_{N-1}=0$. Thus, we solve for the RTO-optimal decision D^*_{N-1} at stage (N-1) to obtain the scalar decision

$$D^*_{N-1} = -\frac{b(ax_{N-1}+fhz_{N-1})}{1+b^2}, \tag{4.8}$$

and, by substituting (4.8) into (4.6), the "optimal" value $\tilde{x}^*_T = f(\tilde{x}_{N-1})$ is obtained, as illustrated below.

Next, we move one additional backward time-step to the new "starting-time" k = (N-2) and repeat the procedure and assumptions just described to solve for the RTO-optimal decision at (N-2), for *arbitrary* values of the composite state $\tilde{x}_{N-2}=(x_{N-2} \mid z_{N-2})$. Thus, $\tilde{J}_{N-2}$ becomes

$$\begin{aligned}\tilde{J}_{N-2} &= \min_{D_{N-2}}\left\{\begin{pmatrix}x_T & z_T\end{pmatrix}\begin{bmatrix}1 & 0\\ 0 & 0\end{bmatrix}\begin{pmatrix}x_T\\ z_T\end{pmatrix}+D^2_{N-1}+D^2_{N-2}\right\}\\ &= \min_{D_{N-2}}\left\{x_T^2+D^2_{N-1}+D^2_{N-2}\right\}.\end{aligned} \tag{4.9}$$

Substituting x_T as given by (4.6) into (4.9) gives

$$\tilde{J}_{N-2} = \min_{D_{N-2}}\left\{\left(ax_{N-1}+fhz_{N-1}+bD_{N-1}\right)^2+D^2_{N-1}+D^2_{N-2}\right\} . \tag{4.10}$$

Substituting the previously computed D_{N-1} as given by (4.8), x_{N-1} as given by (4.4) and observing that the absolute minimum of $\tilde{J}_{N-2}$ with respect to the decision variable D_{N-2} is well defined, exists and is determined by $d\tilde{J}_{N-2}/dD_{N-2}=0$, we solve for the scalar RTO-optimal decision at stage (N-2) to obtain

$$D_{N-2}^{*} = -\frac{ab\left[a^2 x_{N-2} + afhz_{N-2} + fhz_{N-1}\right]}{1+b^2+(ab)^2}. \qquad (4.11)$$

Notice that in (4.11), D_{N-2} is in terms of the variables ($x_{N-2}, z_{N-2}, z_{N-1}$). However, according to (4.3) and (4.4), the instantaneous value of the state z_{N-1} of the structured-variation type exogenous-input is a function of z_{N-2}; i.e.,

$$z_{N-1} = gz_{N-2}, \qquad (4.12)$$

where the $\overline{\sigma}_k$ are being disregarded for the RTO-reasons outlined below (4.4). Thus, D_{N-2} becomes

$$D_{N-2}^{*} = -\frac{ab\left[a^2 x_{N-2} + afhz_{N-2} + fhgz_{N-2}\right]}{1+b^2+(ab)^2}. \qquad (4.13)$$

We continue moving one additional backward time-step and solving for D_k* until we reach $k = t_o$; in this way, we obtain the set of explicit expressions for the RTO *real-time* optimal decisions $D_k^* = D_k^*(\tilde{x}_k)$ with $k=(N-1), (N-2), \ldots, t_o$. Moreover, at each $t_k=k$, the optimal decisions are expressed in terms of the *instantaneous value* of the process state x_k and the *instantaneous value* of the structured-variation exogenous-input "exostate" z_k, both values of which will be available to the decision-maker at each $t_k=k$ by virtue of a real-time "composite-state" observer or Kalman-filter, as described in [35].

4.2 General Expression for the RTO-Optimal Decision for an *N*-Stage, First-Order SDP

The RTO-optimal decision $D_k{*}= D_k{*}(x_k, z_k; k)$ for the *N*-stage first-order sequential-decision process of (4.1)-(4.3) can be generalized to obtain the following scalar expression, which is derived in detail in Appendix A of this dissertation

$$D_k^* = -\left[\frac{a^{(N-1-k)}b}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)}b^2}\right] \times \left[a^{(N-k)}x_k + \sum_{j=k}^{N-1} fh\left(a^{(N-1-j)}\right)z_j\right]. \quad k = 0,1,2,\ldots,N-1, \quad (4.14)$$

where a is the known scalar process state coefficient in (4.1), b is the known scalar decision coefficient in (4.1), f is the known exogenous-input coefficient also in (4.1), and h is the known coefficient of z_k in the exogenous-input state-model (4.3).

It is important to reiterate that the "optimal" decisions here are based-on the RTO principle where, as already mentioned, *at each stage* we tacitly assume the $c_j(k) \equiv constant$ in (3.26) for all "future" times $t > t_k$, or equivalently $\bar{\sigma}_k \equiv 0$ in (3.27), for *all subsequent future times* $t > t_k$. However, when the unknown, unpredictable (but inevitable) time-sparse $\bar{\sigma}_k$ arrive at some unknown future time t, the RTO decisions are *still* based on the instantaneous value z_k of the state of the exogenous-input w_k that is estimated (by the real-time state-observer) at the time t_k and thus, each subsequent decision $D_k{*}$ reflects ("optimally") any changes in the $c_j(k)$ that have occurred since the last optimal decision. This novel strategy achieves RTO-optimal sequential-decisions that are consistently updated at each decision-time, to compensate-for (to the extent possible) any time-sparse $c_j(k)$ changes that may have occurred since the last decision-time. That repeated

corrective-effort achieves "optimal" decisions, in the RTO sense, and thus avoids "treacherous gambling" about time-sparse, non-Ergodic events (the time-sparse sequences of $\overline{\sigma}$ Kronecker delta functions that may, *or may not* "arrive" in the future). Johnson has conjectured [38] that this RTO method of arriving at "optimal" real-time decisions, in the face of unknown, time-sparse changes in the $c_j(k)$-values, corresponds-to the pursuit "decisions" used by humans in competitive-sports like football, basketball, etc., and by animals and other forms of living-things in "predator-prey" processes.

Chapter 5

THE UTILITY OF STRUCTURED-VARIATION TYPE EXOGENOUS-INPUTS IN ENHANCING THE OPTIMALITY OF SEQUENTIAL-DECISIONS IN SDP PROBLEMS

Due to their nature, uncertain exogenous-inputs normally carry a negative connotation in the sense that they are assumed to cause only unwanted, disruptive, non-useful effects on the behavior of the underlying sequential-decision process. However, there are a surprising number of situations in which the exogenous-inputs are capable of producing desirable effects on the underlying process. This notion of the "positive utility" of an exogenous-input that is of the structured-variation type was first introduced by Johnson in [32] for the discrete-time control of continuous-time missiles and other objects of aerospace-type, involving Newtonian dynamical systems. In [32], the "positive utility" of a structured-variation-type exogenous-input was reflected-in the enhancement of the "optimality" of the decisions as revealed in the "better" value of the optimization-criterion. In this chapter, we explore the possible relevance of "utility" of an exogenous-input in sequential-decision problems. In particular, we investigate the detection and quantification of the "usefulness" of an exogenous-input in optimal

sequential-decision processes, when the uncertain time-behavior of the exogenous-input is of the structured-variation type modeled by a state-model of the form (3.27).

5.1 "Utility" of an Exogenous-Input

The idea of the "utility" of an exogenous-input was first introduced in [32] in connection with the optimal disturbance-accommodation control of continuous-time dynamic-systems. The application of that idea to the digital-control of aerospace systems was introduced in [35] in the context-of the theory of Disturbance-Accommodating Control (DAC). In this section, we will give a brief review of the concept of utility as introduced in [35] and then, in the next section, we will extend the results in [35] to SDP with structured-variation-type exogenous-inputs.

In [35], it was shown that the optimal utilization of a structured-variation exogenous-input w_k is achieved by constructing the performance-index (i.e., criterion of optimality J) such that the minimization of J by D_k achieves the primary goal of the process while simultaneously encouraging the "decisions" D_k to make "smart" maximum "use" of the "disruptive effects" of w_k to further optimize J, when that is possible. Thus, as in [35], it is convenient to express the scalar optimization criterion in the following linear-quadratic form

$$J = \frac{1}{2} x_N^T S x_N + \frac{1}{2} \sum_{k=0}^{N-1} \left(x_k^T Q_k x_k + D_k^T R_k D_k \right), \quad \begin{array}{l} x = n\text{-vector} \\ D_k = \text{vector-decision} \end{array}. \qquad (5.1)$$

In (5.1), S, Q, and R are positive definite, symmetric matrices chosen by the designer, N is the specified terminal time, and $(.)^T$ indicates the transpose of (.). The presence of the "decision penalty term $D_k^T R_k D_k$" in the summation of J encourages the maximum

utilization of any "free" energy or process "enhancing effects," available in the exogenous-input w_k on the process, to reduce or minimize the use of <u>excessive</u>, large values of "D_k." This approach has been demonstrated in [35] by invoking the general linearized, discretized dynamic system

$$x_{k+1} = A_k x_k + B_k D_k + F_k w_k \,, \tag{5.2}$$

where A_k, B_k, and F_k are known matrices, x is the process state vector of dimension n, D_k is the decision vector, and w_k is the exogenous-input vector of dimension p. As in (3.27), the discretized dynamic model of the vector exogenous-input uncertain-behavior is assumed to have the general mathematical form

$$\begin{array}{l} w_k = H z_k; \\ z_{k+1} = G z_k + \bar{\sigma}_k \end{array} \; ; \; \begin{array}{l} w_k = p\text{-vector} \\ z_k = \rho\text{-vector} \end{array} , \tag{5.3}$$

where H and G are completely known matrices, the ρ-vector z_k is the state of the exogenous-input vector w_k, and $\bar{\sigma}_k$ is a vector sequence of unknown, time-sparse Kronecker delta functions with completely unknown arrival times and intensities.

In [35], the minimum possible value of the performance criterion J is obtained by using the augmented (composite) state notation $\tilde{x}_k = col.(x_k \mid z_k)$ to endogenize w_k and consolidate the dynamic system into one composite-state model as follows

$$\tilde{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \underbrace{\begin{bmatrix} A_k & F_k H_k \\ 0 & G_k \end{bmatrix}}_{\bar{\mathrm{A}}} \cdot \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \underbrace{\begin{bmatrix} B_k \\ 0 \end{bmatrix}}_{\bar{\mathrm{B}}} \cdot D_k + \underbrace{\begin{bmatrix} 0 \\ \bar{\sigma}_k \end{bmatrix}}_{\bar{\sigma}}, \;\; \tilde{x} \underline{\underline{\Delta}} \left(\frac{x}{z} \right), \tag{5.4}$$

where the meaning of $\bar{\mathrm{A}}$, $\bar{\mathrm{B}}$ and $\bar{\sigma}$ is clear from (5.4).

The minimization of the scalar performance criterion J subject-to the composite dynamic system (5.4) is accomplished by applying conventional discrete-time linear quadratic optimization theory, developed by Kalman [41], using the backward-time solution of the Riccati difference equation and disregarding the time-sparse sequence $\overline{\sigma}_k$ (standard procedure in RTO and DAC theory). That solution leads to the following optimal discrete-time "exogenous-input utilizing" decision [35]

$$D_k^* = -[R_k + \overline{\mathrm{B}}_k^T \overline{\mathrm{P}}_{k+1} \overline{\mathrm{B}}_k]^{-1} \cdot [\overline{\mathrm{B}}_k^T \overline{\mathrm{P}}_{k+1} \overline{\mathrm{A}}_k] \cdot \tilde{x}_k;\; k = N, (N-1), (N-2), \ldots, t_0, \quad (5.5)$$

where $\tilde{x}_k = col.(x_k \mid z_k)$ and the matrix $\overline{\mathrm{P}}$ is the symmetric, positive definite matrix governed by the following Riccati matric difference equation, expressed in "backward-time,"

$$\overline{\mathrm{P}}_k = [\overline{\mathrm{A}}_k^T \overline{\mathrm{P}}_{k+1} \overline{\mathrm{A}}_k + Q_k] - [\overline{\mathrm{B}}_k^T \overline{\mathrm{P}}_{k+1} \overline{\mathrm{A}}_k]^T \cdot [R_k + \overline{\mathrm{B}}_k^T \overline{\mathrm{P}}_{k+1} \overline{\mathrm{B}}_k]^{-1} \cdot [\overline{\mathrm{B}}_k^T \overline{\mathrm{P}}_{k+1} \overline{\mathrm{A}}_k], \quad (5.6)$$

with the known boundary condition (at k=N)

$$\overline{\mathrm{P}}_N = \overline{S};\; N\text{=terminal time}, \quad (5.7)$$

where $\overline{S}$ is the composite weighting matrix on the state at the terminal time with dimension $(n+\rho)\times(n+\rho)$ and defined by $\overline{S} \underline{\underline{\Delta}} \overline{C}^T S \overline{C}$, where $\overline{C}$ is the composite dynamic-system output matrix [35] (recall that p is the dimension of the exogenous-input vector w and ρ is the dimension of the exogenous-input exostate z).

The Riccati difference equation (5.6) is solved by backward-time solution starting at k=(N-1) and moving backwards to k=t_0 to obtain a sequence of values of $\overline{\mathrm{P}}_k$ which can be calculated offline and stored for use in the forward-time decision expression (5.5).

Note that in the decision expression (5.5), the current values of $\overline{\mathrm{A}}_k$ and $\overline{\mathrm{B}}_k$ are used while the "one-step-ahead" value of $\overline{\mathrm{P}}_{k+1}$ is utilized as explained in [35].

It is convenient to partition $\overline{\mathrm{P}}_k$ and $\overline{S}$ into smaller blocks to obtain insight into the structure of the optimal exogenous-input utilizing decision as follows:

$$\overline{\mathrm{P}}_k = \begin{bmatrix} [Kx] & [Kxz] \\ [Kxz]^T & [Kz] \end{bmatrix}, \tag{5.8}$$

where Kx is an $n \times n$ matrix, Kxz is an $n \times \rho$ matrix, and Kz is a $\rho \times \rho$. Substituting (5.8) into the expression (5.5), the decision-law is obtained as

$$D_k^* = -[R_k + B_k^T [Kx]_{k+1} B_k]^{-1} B_k^T \times [[Kx]_{k+1} A_k x_k + ([Kx]_{k+1}(FH)_k + [Kxz]_{k+1} G_k) z_k]. \tag{5.9}$$

The four block matrices in (5.8) obey the following set of matrix difference equations and must be solved offline in backward time k=(N-1), (N-2),…, 2, 1, t_0 using the indicated "initial" conditions at the "end-time" k=N

$$[Kx]_k = \left[A_k - B_k [R_k + B_k^T [Kx]_{k+1} B_k]^{-1} \times B_k^T [Kx]_{k+1} A_k\right]^T [Kx]_{k+1} A_k; \; [Kx]_N = S, \tag{5.10a}$$

$$\begin{gathered} [Kxz]_{k+1} = \left[A_k - B_k [R_k + B_k^T [Kx]_{k+1} B_k]^{-1} \times B_k^T [Kx]_{k+1} A_k\right]^T \left[[Kxz]_{k+1} G_k + [Kx]_{k+1} (FH)_k\right]; \\ [Kxz]_N = 0, \end{gathered} \tag{5.10b}$$

$$\begin{aligned} [Kz]_k = {} & (FH)_k^T [[Kxz]_{k+1} G_k + [Kx]_{k+1} (FH)_k] + G_k^T [[Kxz]_{k+1}^T (FH)_k + [Kz]_{k+1} G_k] \\ & - [[Kx]_{k+1}(FH)_k + [Kxz]_{k+1} G_k]^T \times B_k [R_k + B_k^T [Kx]_{k+1} B_k]^{-1} \\ & \times B_k^T [[Kx]_{k+1} (FH)_k + [Kxz]_{k+1} G_k]; \; [Kz]_N = 0. \end{aligned} \tag{5.10c}$$

The "optimal" value of the optimization criterion J when the optimal decision (5.9) is used is called the "value"-function and is denoted by the scalar function

$\mathcal{V}(x, z, k)$ and it represents the absolute minimum value of J obtained by making the unconditional optimal decision D_k at each k. Kalman has shown [41] that the "value-function" $\mathcal{V}(x, z, k)$ [50] which is the value of J using the optimal decision D_k^* (i.e., the minimum possible value of J) has the "quadratic form" structure [50]

$$\mathcal{V} = \frac{1}{2}\tilde{x}_k^T \overline{\mathrm{P}}_{k+1}\tilde{x}_k = \frac{1}{2}(x|z)^T \times \begin{bmatrix} [Kx] & [Kxz] \\ [Kxz]^T & [Kz] \end{bmatrix} \begin{pmatrix} x \\ z \end{pmatrix}. \tag{5.11}$$

Expanding (5.11) yields

$$\mathcal{V} = \frac{1}{2}\overbrace{\left(x_x^T[Kx]_{k+1}x_k\right)}^{\text{Fixed Cost}} + \overbrace{\left(x_x^T[Kxz]_{k+1}z_k\right)}^{\text{-Assistance}} + \frac{1}{2}\overbrace{\left(z_k^T[Kz]_{k+1}z_k\right)}^{\text{Burden}}, \tag{5.12}$$

where the terms "fixed cost," "assistance" and "burden" indicate the effect of the state x and the exostate z on the "optimal"-value of the optimization criterion J. Since the objective is to <u>minimize</u> J, and the matrix Kz is positive definite for all k, the last term in (5.12) is always positive and represents an unavoidable increase in $\mathcal{V}$ due to the sheer presence of the exogenous input $w_k = Hz_k$. This is why that term is identified as the "burden" of the exogenous-input [35]. The second term in (5.12) is a bi-linear term (Kxz is <u>not</u> square, in general) having <u>no</u> sign-definite properties, and represents the contribution to J from the interaction of x_k and z_k. Since the net sign of that second term can be either positive, negative or zero, any usefulness (reduction in the value of $\mathcal{V}$) from the presence of the exogenous-input w_k must occur through the negative value of this term. Thus, the negative of this term is called the "Assistance" contributed by z_k [35]. The first term in (5.12) is identified as the "fixed cost" since its contribution to the value of $\mathcal{V}$ is independent of the exogenous-input z_k.

The utility $\mathcal{U}$ of the exogenous-input is defined [35] as the difference between the value of $\mathcal{V}$ when $w_k \equiv 0 \Rightarrow z_k \equiv 0$ and the value of $\mathcal{V}$ when $w_k \neq 0$ $(z_k \neq 0)$. Thus, the utility $\mathcal{U}$ of w_k can be expressed as $\mathcal{U}$ = Assistance of w_k – Burden of w_k as follows:

$$\mathscr{U}_k = -\left(x_k^T [Kxz]_{k+1} z_k\right) - \frac{1}{2}\left(z_k^T [Kz]_{k+1} z_k\right). \tag{5.13}$$

From (5.13), it can be observed that if $\mathcal{U}$>0, the net effect of the exogenous-input w_k on the value $\mathcal{V}$ of the optimal performance is to help by reducing the value of $\mathcal{V}$ compared to the value when $w_k \equiv 0$. On the other hand, if $\mathcal{U}$<0, the exogenous-input w_k is unavoidably increasing the value of $\mathcal{V}$. But, even in that case, the decision D_k nevertheless "optimally-minimizes" the unavoidable increase in J which is a win-win outcome for the SDP! Expression (5.13) provides a real-time assessment of the instantaneous utility of the exogenous-input w_k at time $k, t_0 < k < N$, in the RTO sense. In particular, it determines, in the RTO sense, the sign and magnitude of the "help" (utility) which the exogenous-input will contribute over the remaining decision interval $k < k' < N$, assuming $\sigma_k \equiv 0$ over that remaining interval. This assessment is accurate until the next unpredictable arrival of a "surprise" Kronecker function σ_k, at which point the net effect of all intervening unpredictable "jumps" in the state of w_k will be evaluated and a re-evaluation of $\mathcal{U}$=$\mathcal{U}(x, z)$ will occur, as in all RTO decision-processes.

5.2 Derivation of the Utility Expression $\mathcal{U} = \mathcal{U}(x,z)$ for a First-Order Linear SDP Example with Linear Quadratic-type Optimality Criterion *J* and Structured-Variation Type Uncertain Exogenous-Inputs

In this section, a specific example is considered and the Dynamic Programming solution methodology for determining optimal SDP decisions is employed to determine the useful effects (if any) of an exogenous-input w_k that is of the structured-variation type. Recall that the minimum possible (optimal) value of *J* for the "exogenous-input utilizing (EIU)" decision D_k was defined by the scalar function $\mathcal{V}$ in (5.12). Continuing with the first-order SDP example of Section 4.1, the first step is to determine the scalar function $\mathcal{V}$. Since the optimal "exogenous-input-utilizing" decision D_k* has already been found for the first order sequential decision process of (4.1)-(4.3) and is given by (4.14), it is not necessary to solve the Riccati difference equation (5.6) for this case.

The scalar function $\mathcal{V}$ for the first-order SDP of (4.1)-(4.3) is the minimum possible value of *J* obtained with the Dynamic Programming solution method when structured-variation exogenous-inputs are present and the RTO-optimal decision (4.14) is employed. For this purpose, we compute the optimization criterion (5.1) using the RTO decision (4.14) at each decision stage starting with (N-1). For this case, we will assume R=1, Q=0, and S=1 (note that this results in the same optimization criterion of (4.2)). Thus, the minimum value $\mathcal{V}_{RTO}$ of the RTO optimization criterion at decision stages (N-1), (N -2), (N -3), and (N -4) is found from (5.1) to be

$$\mathcal{V}_{\text{RTO(N-1)}} = (a^2 x_{N-1}^2 + 2afhx_{N-1}z_{N-1} + f^2h^2z_{N-1}^2)\big/(1+b^2)$$
$$\mathcal{V}_{\text{RTO(N-2)}} = (a^4 x_{N-2}^2 + 2a^2 fh(a+g)x_{N-2}z_{N-2} + f^2h^2(a+g)^2 z_{N-2}^2)\big/(1+b^2+a^2b^2)$$
$$\mathcal{V}_{\text{RTO(N-3)}} = \frac{(a^6 x_{N-3}^2 + 2a^3 fh(a^2+ag+g^2)x_{N-3}z_{N-3} + f^2h^2(a^2+ag+g^2)^2 z_{N-3}^2)}{(1+b^2+a^2b^2+a^4b^2)}$$
$$\mathcal{V}_{\text{RTO(N-4)}} = \frac{(a^8 x_{N-4}^2 + 2a^4 fh(a^3+a^2g+ag^2+g^3)x_{N-4}z_{N-4} + f^2h^2(a^3+a^2g+ag^2+g^3)^2 z_{N-4}^2)}{(1+b^2+a^2b^2+a^4b^2+a^6b^2)}, \tag{5.14}$$

where a, b and f are the scalar, known, arbitrary real functions as defined in (4.1) and h and g are real, known, constant coefficients defined in (4.3) as part of the exogenous-input exostate model.

It is shown in Appendix B that the results (5.14) can be generalized to the following scalar expression [21]

$$\mathcal{V}_{RTO}(k) = \left(\frac{a^{2(N-k)}}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)}b^2} \right) x_k^2 + 2\left(\frac{a^{(N-k)} fh\overline{(a+g)}^{(N-k)}}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)}b^2} \right) x_k z_k + \left(\frac{(fh)^2 \overline{(a+g)}^{2(N-k)}}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)}b^2} \right) z_k^2, \tag{5.15}$$

where, as shown in Appendix B, the term $\overline{(a+g)}^{(N-k)}$ is defined as a modified form of the Binomial theorem in which the coefficients of the interior terms are all equal to one. Thus,

$$\overline{(a+g)}^n \triangleq a^n + \frac{n}{n}a^{n-1}g + \ldots + \frac{{}_nC_m}{{}_nC_m}a^{n-m}g^m + \ldots + \frac{n}{n}ag^{n-1} + g^n, \tag{5.16}$$

where the Binomial coefficient of $a^{n-m}g^m$ is given by [45]

$$
{}_{n}C_{m} = \frac{n!}{m!(n-m)!} = \frac{n(n-1)(n-2)\ldots(n-m+1)}{m!}, \tag{5.17}
$$

where n and m are positive integers.

By inspecting the structure of (5.15), it is possible to decompose (5.15) and identify the "fixed cost", "assistance" and "burden" terms as follows:

$$
\mathcal{V}_{RTO}(k) = \overbrace{\left(\frac{a^{2(N-k)}}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)} b^2} \right) x_k^2}^{\text{Fixed Cost}} + \overbrace{2\left(\frac{a^{(N-k)} fh \overline{(a+g)}^{(N-k)}}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)} b^2} \right) x_k z_k}^{\text{-Assistance}} + \overbrace{\left(\frac{(fh)^2 \overline{(a+g)}^{2(N-k)}}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)} b^2} \right) z_k^2}^{\text{Burden}}. \tag{5.18}
$$

As mentioned in the previous section, the utility of the exogenous-input is defined as the *difference* between the value of $\mathcal{V}$ when $w_k \equiv 0$ and the value of $\mathcal{V}$ when $w_k \neq 0$. Similarly, the expression for the utility $\mathcal{U}_{RTO}$ of w_k can be obtained as in (5.13) by using the definition $\mathcal{U}_{RTO} \triangleq$ Assistance – Burden, as follows [21]:

$$
\mathcal{U}_{RTO}(k) = -2\left(\frac{a^{(N-k)} fh \overline{(a+g)}^{(N-k)}}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)} b^2} \right) x_k z_k - \left(\frac{(fh)^2 \overline{(a+g)}^{2(N-k)}}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)} b^2} \right) z_k^2. \tag{5.19}
$$

As in the general utility expression (5.13), expression (5.19) provides a real-time assessment of the instantaneous utility, in the RTO sense, of the exogenous-input z_k at time *k*, $t_0 < k < T$, determining the sign and magnitude of the help which the exogenous-input will contribute over the remaining decision interval $k < k' < T$. This assessment is

accurate until the next arrival of a time-sparse (unpredictable) “surprise” $\bar{\sigma}_k$ at which point the (unpredictable) “jump” in w_k (and in z_k) will involve a re-evaluation of $\mathcal{U}_{RTO}$.

By rewriting equation (5.19) in terms of only x_k and z_k and not in terms of the exogenous-input exostate term g in (4.3), we obtain the following form for the utility equation [21]

$$\mathscr{U}_{RTO}(k) = -\left[\frac{1}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)}b^2}\right] \times \left[2a^{(N-k)}x_k \sum_{j=k}^{N-1}(a^{(N-1-j)} fhz_j) + \left(\sum_{j=k}^{N-1}(a^{(N-1-j)} fhz_j)\right)^2\right], \quad (5.20)$$

which has a form analogous to the RTO optimal decision D_k* of (4.14).

5.3 Utility Domains in (x, z)-Space

In [35], the boundaries separating the domains of positive and negative utility in the (x, z)-space are defined by the collection of points in the (x, z)-space where utility is zero. For the general SDP case of Section 5.1, the latter domains are defined by

$$\mathscr{U}_k = -\left(x_k^T [Kxz]_{k+1} z_k\right) - \frac{1}{2}\left(z_k^T [Kz]_{k+1} z_k\right) = 0\,. \quad (5.21)$$

Since Kz is a square and non-singular matrix, the set of values (x, z) defined by equation (5.21) are

$$z = 0;\;\; z_k^T = -2x_k^T [Kxz]_{k+1}^T \left[[Kz]_{k+1}\right]^{-1}, \quad (5.22)$$

and equivalently for our specific SDP example (4.1)-(4.3), the domains of utility in the (x, z)-space are defined by

$$\mathscr{U}_{RTO}(k) = -\left[2a^{(N-k)} x_k \sum_{j=k}^{N-1} (a^{(N-1-j)} fhz_j) + \left(\sum_{j=k}^{N-1} (a^{(N-1-j)} fhz_j) \right)^2 \right] = 0\,, \quad (5.23)$$

where a is the known but arbitrary scalar coefficient of the process state described in (4.1).

Due to the geometrical complexities of expressions (5.22) and (5.23), a description of the positive utility regions in (*x, z*)-space is not feasible for the general case. The regions of positive and negative utility vary with time *k* since the gains *Kxz* and *Kz* are time-varying solutions of difference equations. However, for the especial case of the linear first-order SDP of (4.1)-(4.3) where a is a scalar, known coefficient, expression (5.23) allows the determination of the utility domain in (*x-z)*-space as seen in Figure 5.1 (shown in the (**$\mathscr{U}$**-*x-z)*-space) for the values *N*=2*, n*=*ρ*=1*,* and *a*=*f*=*h*=*b*=1.

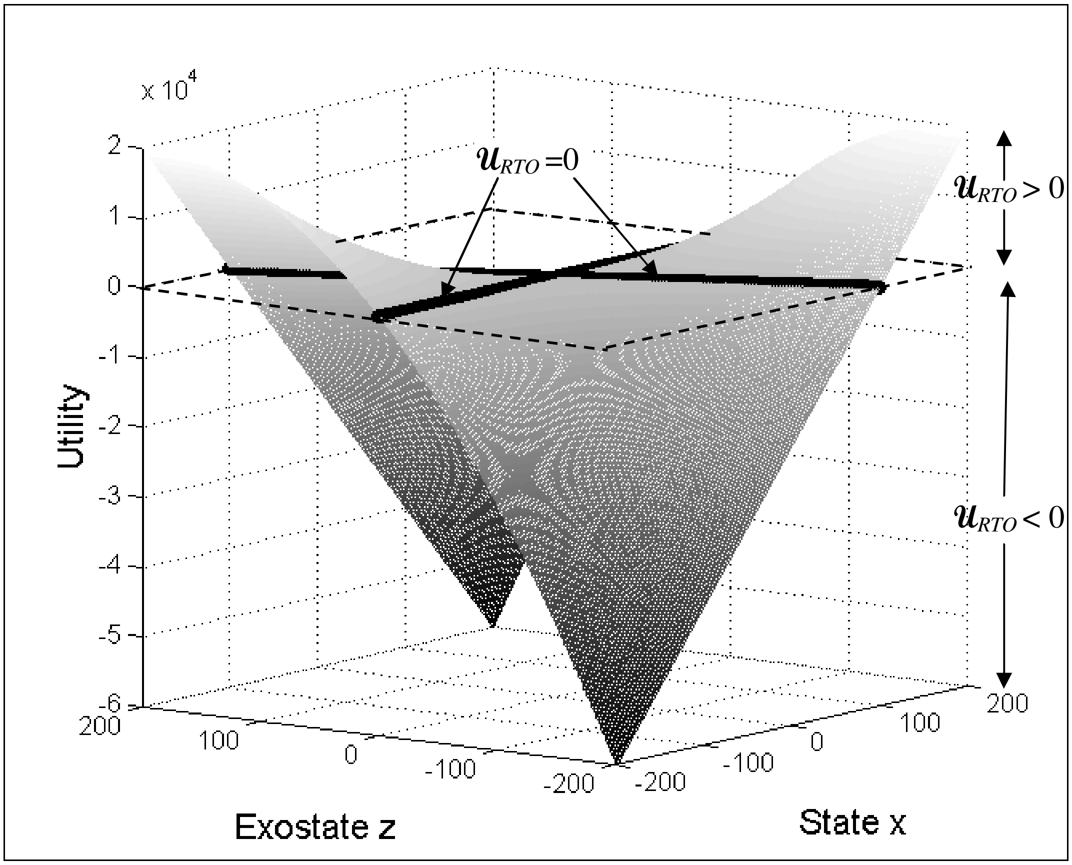

Figure 5.1. Utility Surface and Regions of Positive Utility ($\mathcal{U}$>0), Negative Utility ($\mathcal{U}$<0) and Zero Utility ($\mathcal{U}$=0) for Problem (4.1)-(4.3) with a Structured-Variation Exogenous-Input; N=2, n=1, ρ=1, a=1, b=1, f=1, g=1, h=1.

5.4 Effectiveness of an RTO-Optimal Exogenous-Input Utilizing Decision

It is important to quantify how much better the "exogenous-input utilizing" decision performs when compared to the case in which no exogenous-inputs are present. Kelly [43] proposed the concept of effectiveness $\mathcal{E}$ for a discrete-time linear-quadratic regulator. We will use Kelly's expression modified to work with the sequential decision processes discussed in this dissertation as follows:

$$\mathcal{E} \underline{\underline{\Delta}} \frac{J_{nw} - J_{RTO}}{J_{nw}} \times 100\%, \tag{5.24}$$

where J_{nw} represents the optimum (minimum) value of the optimization criterion J obtained when no exogenous-inputs are present ($w_k \equiv 0; \forall\ k$) and J_{RTO} represents the value of the RTO-optimum (minimum) value of the optimization criterion J obtained when exogenous-inputs of the structured variation-type are acting-on the process. In both cases, the process is subjected to the same initial conditions except for the presence of the uncertain exogenous-input.

When the "exogenous-input utilizing decision" of (4.14) is performing better (that is achieving a smaller value of J), the effectiveness $\mathcal{E}$ in (5.24) is *positive* and the larger the (positive) value of $\mathcal{E}$, the greater the effectiveness of the "exogenous-input utilizing (EIU)" decision. The maximum possible value of $\mathcal{E}$ is 100% which could only occur if $J_{RTO} = 0$ (very unlikely). Note that in general, the value of $\mathcal{E}$ can also be *negative*.

Chapter 6

SOME ILLUSTRATIVE EXAMPLES

Using MATLAB, optimization and utility studies were performed on a set of illustrative examples applying the more-general RTO Principle and the corresponding Dynamic Programming methodology developed here, for the case of structured-variation type exogenous-inputs. Appendix C shows the MATLAB code used for the examples illustrated in this chapter. For all the examples in this chapter, the sequential-decision process is assumed to be governed by the following first-order, linear state evolution equation

$$x_{k+1} = ax_k + bD_k + fw_k \ , \tag{6.1}$$

where the discrete integer "stages" k in the sequential-decision problem extend over the time interval $0=t_0 \leq k \leq 4 = T$ (T =terminal time), and the coefficients (a, b, and f) are scalar and known constants (*time-invariant*). To simplify the calculations, we will consider the case in which a, b, and f are all equal to 1 except when generalizing an equation.

The key to achieving maximum utilization of exogenous-inputs in a sequential decision process is in the selection of a criterion of optimality J. In particular, one should

design the mathematical structure of J so that when J is minimized with respect to the decision variable, the primary decision objective is achieved and, at the same time, maximum use of the exogenous-input w_k is mathematically "encouraged." It turns out that, for a broad family of realistic SDP's, one can accomplish this design-goal by selecting the criterion of optimality illustrated generically in (4.2), which is a modified form of (5.1) and is repeated here for convenience

$$J = {x_T}^2 + \sum_{k=0}^{N-1} (D_k^2) \,. \tag{6.2}$$

Just as in (3.2), the first term in the optimization expression (6.2) indicates penalization of deviations of the state from the desired equilibrium while the second term discourages excessive use of the decision effort and at the same time utilizes any "free" energy available in the exogenous-input w_k [32]. This form of optimization criteria yields an RTO-optimal control which uses the exogenous-input w_k to assist in minimizing/maximizing the optimization criteria J whenever possible. When such assistance is not available from the exogenous-input, this type of RTO-optimal controller *still* minimizes the inevitable increase in the optimization criteria J caused by the exogenous-input w_k.

For all the following examples, it is also assumed that a discrete-time "composite-state" observer or Kalman Filter is employed [35]. This observer simultaneously produces on-line, *real-time* estimates of the process state x_k and the exogenous-input exostate z_k at *each* decision stage k. That real-time estimate of $(x|z)$ embodies the maximum-information "knowable" at each decision-time t_k and thus is the information needed to make a rational decision that is RTO-optimal. Since the purpose of the present

work is to study the theoretically "best-possible" utility of the exogenous-input under idealized implementation-conditions, it is assumed that the composite-state estimator is ideal and thus provides "perfect" real-time estimates of the process and exogenous-input's states (x_k, z_k) at each decision-time k.

Processes of the form (6.1) and (6.2) arise from Dynamic Programming solutions of SDP problems in economics, demographics, business, etc., and are intrinsically discrete-in-time. Thus, the mathematical models for the processes of the form (6.1) presented here are expressed in discrete form since they do not necessarily have an underlying continuous counterpart while, the exogenous-inputs that act on those processes are also discrete-time in nature.

6.1 Example #1: The Case of Zero Exogenous-Input (Baseline Case)

This is the baseline case which will be used as a basis for utility comparisons of all cases presented later. It is assumed that the uncertain exogenous-input in this Example does not influence the process because its value is zero. Thus, $w(t_k)$ can be represented by the trivial discrete-time difference equation

$$w_{k+1} = w_k ; w_0 \underline{\underline{\Delta}} 0 \quad 0 \le k \le 4 . \tag{6.3}$$

The associated discrete state-space model of the exogenous-input w_k can be obtained by defining $z_k \underline{\underline{\Delta}} w_k$ and is as follows:

$$\begin{aligned} & w_k = 0; \\ & z_{k+1} = z_k ; z_0 = 0. \end{aligned} \tag{6.4}$$

In <u>forward</u>-time and starting at t_0=0, we apply the previously computed backward-time optimal decisions of (4.14) which were determined by normal Dynamic Programming

calculations. Thus, the backward-time sequence of RTO-optimal decisions obtained from (4.14), $D_k^*(x_k, z_k)$ with k=3, 2, 1, 0 and z_k =0 are also strictly (absolutely) optimal decisions and are

$$D_3^* = -(x_3 + z_3)/2 = -x_3/2 \ . \tag{6.5}$$

$$D_2^* = -(x_2 + z_2 + z_3)/3 = -x_2/3 \ . \tag{6.6}$$

$$D_1^* = -(x_1 + z_1 + z_2 + z_3)/4 = -x_1/4 \ . \tag{6.7}$$

$$D_0^* = -(x_0 + z_0 + z_1 + z_2 + z_3)/5 = -x_0/5. \tag{6.8}$$

The value of the RTO-optimization criterion J obtained at the end of the process (T=4) in this example is, in fact, the optimal (absolute minimum) value of J which is obtained when there is not an exogenous-input (no "uncertainty" of any kind) acting on the system process and is computed as follows:

$$J_{RTO} = \left\{ (x_T)^2 + \left[-\frac{x_3}{2}\right]^2 + \left[-\frac{x_2}{3}\right]^2 + \left[-\frac{x_1}{4}\right]^2 + \left[-\frac{x_0}{5}\right]^2 \right\} \ , \tag{6.9}$$

where x_T is the state obtained at the terminal time T (not known a priori). The above expression is the same as the Fixed Cost defined in (5.18) and as a result, the utility $\mathcal{U}_{RTO}$ is zero due to the absence of an exogenous-input. By rewriting (6.9) in terms of only the initial state x_0, we obtain the optimization criterion $J(x_0)$ at the end of the process as follows:

$$J_{RTO} = (x_0)^2/5 \, . \tag{6.10}$$

Thus, for example, when x_0=∓20, the absolute minimum value obtainable for the value of the optimization criterion is J=80.

6.2 Example #2: The Case of an Unknown Single-Step, "Stepwise-Constant"-Type Exogenous-Input

The uncertain exogenous-input in this Example #2 can be approximated as an unknown constant-step that jumps in value at one time t^*, where t^* is an unknown. Thus, the uncertain exogenous-input can be represented by the following "spline"-type function

$$w_k = \begin{cases} C_1 & 0 \le k < t_2 \quad C_{1,2} = \text{unknown constants} \\ C_2 & t_2 \le k \le t_4 \quad t_2 = \text{ unknown} \end{cases}, \tag{6.11}$$

which is depicted in Figure 6.1 for the specific case C_1=2 and C_2=0 to be used in Example #2.

The discrete state-space model of the exogenous-input w_k is obtained from the generic state-space model in (4.3) by finding the lowest order homogenous difference equation for which (6.11) is the general solution (see Appendix D) and defining $z_k \triangleq w_k$ and is as follows:

$$\begin{aligned} w_k &= hz_k; \quad h = 1; \\ z_{k+1} &= gz_k + \bar{\sigma}_k; \quad g = 1, \end{aligned} \tag{6.12}$$

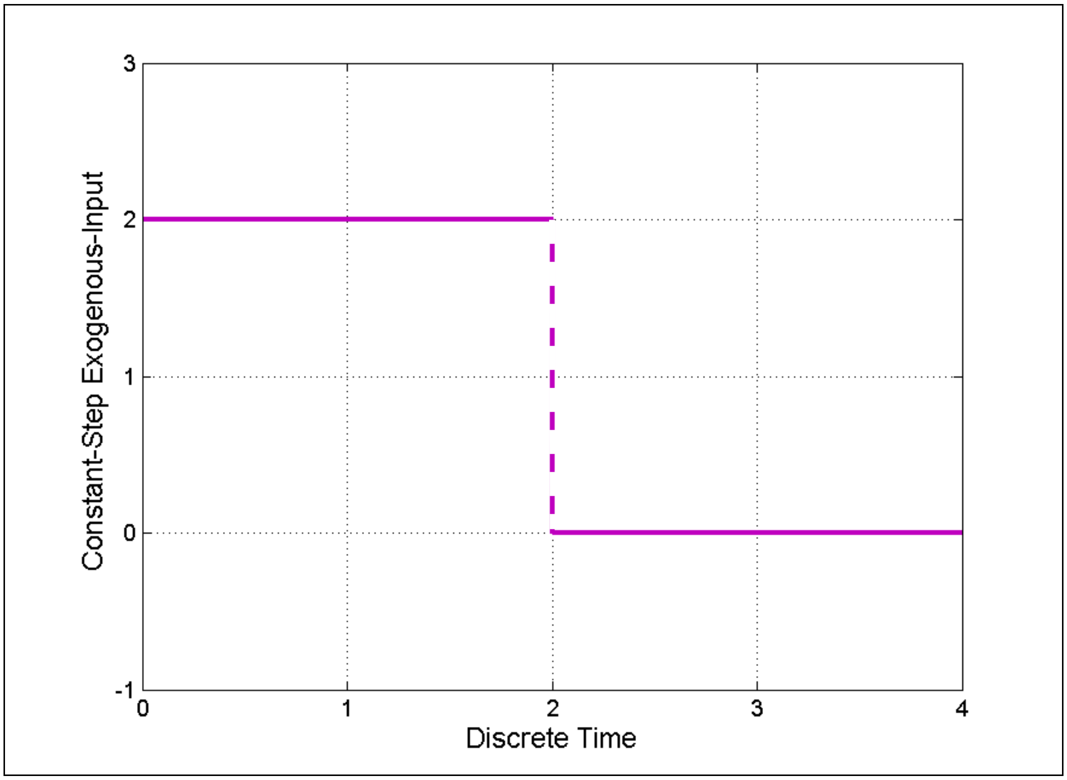

Figure 6.1. Assumed Unknown Constant-Step, Structured-Variation Exogenous-Input for Example #2; C_1=2 and C_2=0

where $\overline{\sigma}$ is a sequence of *time-sparse* Kronecker delta functions with unknown arrival times and values and accounts for the unknown single "jump" of the constant-step in Example #2. The composite state $\tilde{x}$ consisting of the underlying process state x and the structured-variation type exogenous-input "exostate" z, is governed by the following discrete-time state-evolution model

$$\tilde{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \begin{bmatrix} 1 \\ 0 \end{bmatrix} \cdot D_k + \begin{bmatrix} 0 \\ \sigma_k \end{bmatrix}, \quad \tilde{x} \underline{\underline{\Delta}} \left(\frac{x}{z} \right), \tag{6.13}$$

where k=0, 1, 2, 3, and $t_N = T$=4. To solve this problem, we utilize the previously determined optimal decisions obtained in Sections 4.1 and 4.2 with the Dynamic

Programming <u>*backward*</u>-recursive technique and apply the RTO Principle in <u>*forward*</u>-*time*. The backward time RTO-optimal decisions obtained from (4.14) are repeated here and are

$$D_3^* = -(x_3 + z_3)/2. \tag{6.14}$$

$$D_2^* = -(x_2 + z_2 + z_3)/3 \ . \tag{6.15}$$

$$D_1^* = -(x_1 + z_1 + z_2 + z_3)/4. \tag{6.16}$$

$$D_0^* = -(x_0 + z_0 + z_1 + z_2 + z_3)/5\,. \tag{6.17}$$

To apply the RTO principle, we assume an arbitrary <u>known</u> initial exogenous-input value, w_0. We also tacitly assume that the exogenous-input value w_0 will <u>remain unchanged</u> from the current time, k=0, until the end of the process k=4 ($\overline{\sigma} \equiv 0$ assumption). When, in fact, the value of the exogenous-input *changes* at some later time k=2 (although this fact and its timing are completely unknown to the decision maker), the "composite-state" observer determines the real-time value of the "state" z_2 of the actual exogenous-input w_2 at *that* later decision time (k=2) and thus, $\tilde{x}(t_2)$ embodies all the information needed to make a rational, *real-time* decision at time k=2 that is RTO-optimal.

According to (6.12) and (6.13), the values of the exogenous-input at time $k=0$ are defined as

$$
\begin{aligned}
&w_0 = z_0 = C_1; \\
&z_1 = z_0 + \sigma_0 = C_1 + \sigma_0; \\
&z_2 = z_1 + \sigma_1 = C_1 + \sigma_0 + \sigma_1; \\
&z_3 = z_2 + \sigma_2 = C_1 + \sigma_0 + \sigma_1 + \sigma_2,
\end{aligned} \tag{6.18}
$$

where the value of z_0 is the known initial value of the exogenous-input. Since we are applying the RTO principle, as previously mentioned, we are tacitly assuming that the exogenous-input z_0 will remain unchanged from the current time, k=0, until the end of the process at k=4 (the constant C_1 in (6.11) remains unchanged). Thus, at time k=0, $\sigma_0 = \sigma_1 = \sigma_2 = 0$ and the decision D_0 in (6.17) becomes

$$
D_0 = -\frac{(x_0 + 4C_1)}{5}. \tag{6.19}
$$

At time k=1 and according to (6.12), the values of the exogenous-input are determined to be

$$
\begin{aligned}
&w_1 = z_1 = C_1; \\
&z_2 = z_1 + \sigma_1 = C_1 + \sigma_1; \\
&z_3 = z_2 + \sigma_2 = C_1 + \sigma_1 + \sigma_2.
\end{aligned} \tag{6.20}
$$

Since we are applying the RTO Principle, we again are tacitly assuming that the current value of the exogenous- input z_1 will remain unchanged until the end of the process. Thus, $\sigma_1 = \sigma_2 = 0$ and the previously determined RTO-optimal decision D_1 in (6.16) becomes

$$
D_1 = -\frac{(x_1 + 3C_1)}{4}. \tag{6.21}
$$

Similarly, at k=2, according to (6.12), we know that

$$\begin{aligned} w_2 &= z_2 = C_2; \\ z_3 &= z_2 + \sigma_2 = C_2 + \sigma_2, \end{aligned} \tag{6.22}$$

where the "composite-state" observer has estimated the "jump" in the exogenous-input and determined the new value of z to be C_2. Once again, we tacitly assume that the current value of the exogenous-input will remain at this new value from the current time k=2 until the end of the process. Thus, $\sigma_2 = 0$ and the previously determined RTO-optimal decision D_2 in (6.15) becomes

$$D_2 = -(x_2 + 2C_2)/3\,. \tag{6.23}$$

Finally, at k=3, $w_3 = z_3 = C_2$ and the RTO-optimal decision (6.14) becomes

$$D_3 = -(x_3 + C_2)/2\,. \tag{6.24}$$

At the end of the process at k=4, we obtain the RTO-minimized optimization criteria as

$$J = \left\{ (x_T)^2 + \left[-\frac{(x_3 + C_2)}{2} \right]^2 + \left[-\frac{(x_2 + 2C_2)}{3} \right]^2 + \left[-\frac{(x_1 + 3C_1)}{4} \right]^2 + \left[-\frac{(x_0 + 4C_1)}{5} \right]^2 \right\}, \tag{6.25}$$

in which the terminal state x_T is the optimal state obtained at the end of the process at k=4.

Figure 6.2 illustrates the value of the optimization criterion (6.25) for the constant-step of (6.11) with three different initial state conditions x_0. The lines in Figure 6.2 were obtained by varying C_1 with $-20 \le C_1 \le 20$ and letting C_2=0. The plus

(+) line represents J for the case in which x_0=0, the circle (o) line is for the case x_0=+20, and the dotted line is the case x_0=-20.

Figure 6.2 also shows the optimization criterion value of the previous example. Thus, the gray dashed line represents the value of J for the baseline case without an exogenous-input, as defined in (6.10) for two initial condition cases in which x_0=±20. Figure 6.2 clearly shows the "utility" of the constant-step exogenous input when compared to the baseline case (zero exogenous-input). By comparing the value of the optimization criterion J when there is not an exogenous-input present (i.e., $w=0$ and $J=80$) to the value of J when the exogenous-input is a constant-step and with x_0=±20, one can observe that the J is minimized further when the constant-step (and in particular when C_1) takes any value in the open interval (0, 12) for x_0=-20 and when C_1 takes any value in the open interval (-12, 0) for x_0=+20. When the exogenous-input is not present, the minimum possible value of J is *80 for* x_0=±20. On the other hand, when the exogenous-input is a constant-step as described by (6.11), the value of J is lowered to less than *80* when C_1 takes certain values. This area of positive utility is depicted by the shaded region in Figure 6.2.

According to (5.19) and (5.20) and using (6.18), (6.20), (6.22) and the RTO principle, the value of the *instantaneous utility* $\mathcal{U}_k$ at each discrete time k is as follows

$$\mathcal{U}_0 = -\frac{(z_0+z_1+z_2+z_3)^2 + 2x_0(z_0+z_1+z_2+z_3)}{5} = -\frac{8}{5}\left(x_0C_1 + 2C_1^2\right), \tag{6.26}$$

$$\mathcal{U}_1 = -\frac{(z_1+z_2+z_3)^2 + 2x_1(z_1+z_2+z_3)}{4} = -\frac{3}{4}(2x_1C_1 + 3C_1^2), \tag{6.27}$$

$$\mathcal{U}_2 = -\frac{(z_2+z_3)^2 + 2x_2(z_2+z_3)}{3} = -\frac{4}{3}(x_2C_2 + C_2^2), \tag{6.28}$$

$$\mathscr{U}_3 = -\frac{z_3(2x_3 + z_3)}{2} = -\frac{1}{2}(2x_3C_2 + C_2^2)\,. \tag{6.29}$$

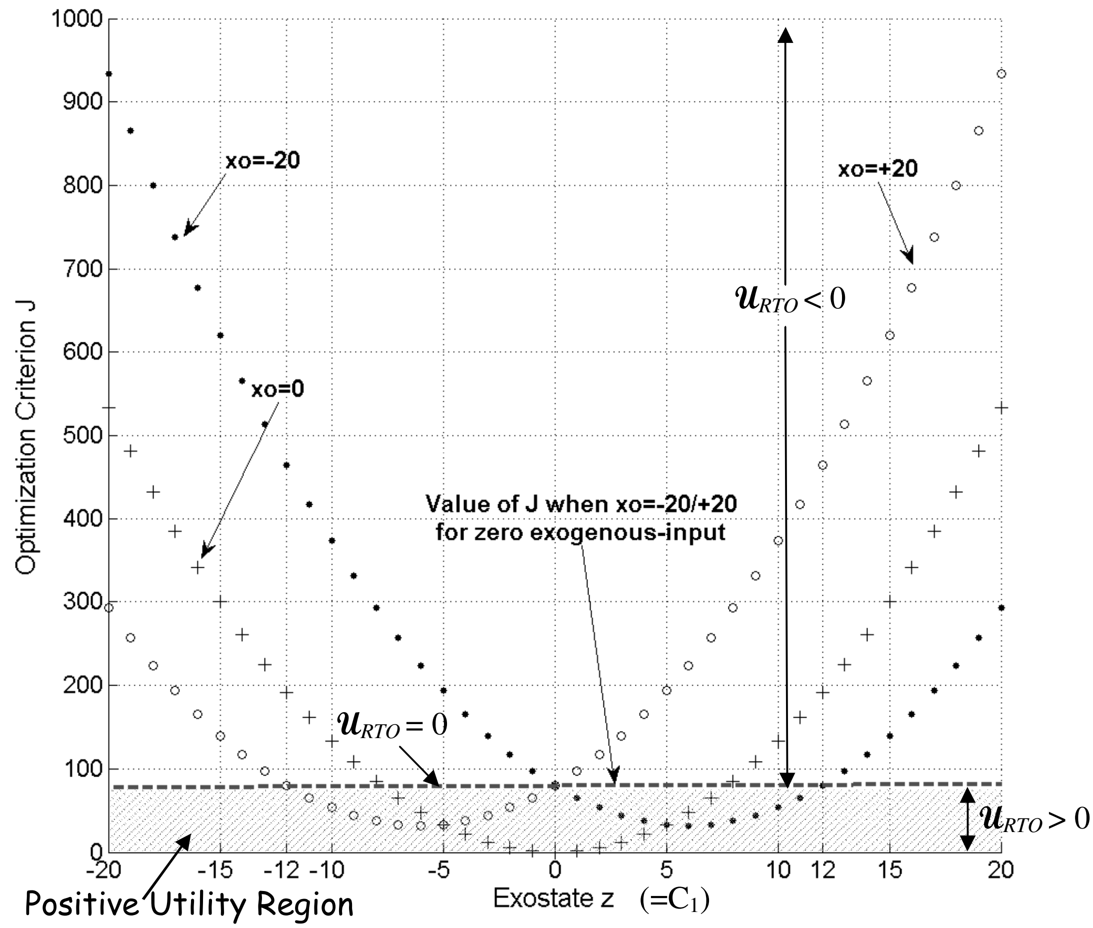

Figure 6.2. Exogenous-Input Utility and Optimal Values of J=J(x_0) for Zero and Constant-Step Exogenous-Inputs; Calculated from (6.25)

In Figure 6.3, four utility surfaces encompassing positive, negative and zero utility are shown, one for each discrete time k for which a decision is made. These utility surfaces are for the most general case of (6.1), (6.2), and (6.11) and were obtained using

the utility expression (5.20) for $-40 \le x \le 40$ and $-40 \le z \le 40$. The dotted lines on the planes are the values of the utility in the ($\mathcal{U}$-x-z)-space for this specific example in which the exogenous-input z is a constant-step that takes the values of C_1 and C_2 as described by (6.11) and the state x only takes the values given by (6.1). The general-utility surfaces are shown to illustrate that the utility surfaces for this example fall on the general-utility surfaces (similar to those of Figure 5.1) as expected. The different lines that form the utility tracks (dark dots on the surfaces) were obtained by setting C_1 to a specific constant value and $C_2=0$ for the entire forward Dynamic Programming process k =0, 1, 2, 3 and using the state evolution equation of (6.1) together with the utility expression of (5.20). Then, a new constant-step value (essentially a new C_1 value) was selected and the utility at each decision time computed again. This process was repeated for different values of the constant-step exogenous input constant C_1 with $-20 \le C_1 \le 20$. The dotted lines represent the value of the instantaneous utility for different initial states x_0 as annotated in Figure 6.3.

Figures 6.4 and 6.5 show rotated views of Figure 6.3 without the general utility surfaces. In Figure 6.4, one can observe that the optimal state x_k progressively moves through regions of positive utility $\mathcal{U}_{RTO}>0$ and negative utility $\mathcal{U}_{RTO}<0$ for the cases $x_0=\pm 20$. This demonstrates that <u>if</u> the sequential-decisions are "Exogenous-Input Utilizing (EIU)-smart" decisions, the constant-step exogenous-input of (6.11) contributes to *further minimizing* the optimization criterion in certain regions of the ($\mathcal{U}$-x-z)-space. This is a clear indication that when EIU-smart sequential-decisions are used, certain values of the constant-step exogenous-input of (6.11) acting on the state described by (6.1) and with an optimization criterion as described by (6.2) are in fact "useful" in

further decreasing the value of the optimization criterion *J*. However, that benefit does-not "automatically" occur but, instead, requires extraordinary-clever, real-time "decisions" that can only be conceived by the use of a "smart" exogenous-input utilizing "sequential-decision" algorithm (policy) as illustrated here. As Johnson (the originator of EIU-theory which he called Disturbance Utilizing Control, DUC) has remarked [35], the (control) decisions required-to harvest the useful benefits of disturbance-inputs involve extreme finesse, not unlike competitive sailing against a head-wind by skillful "tacking."

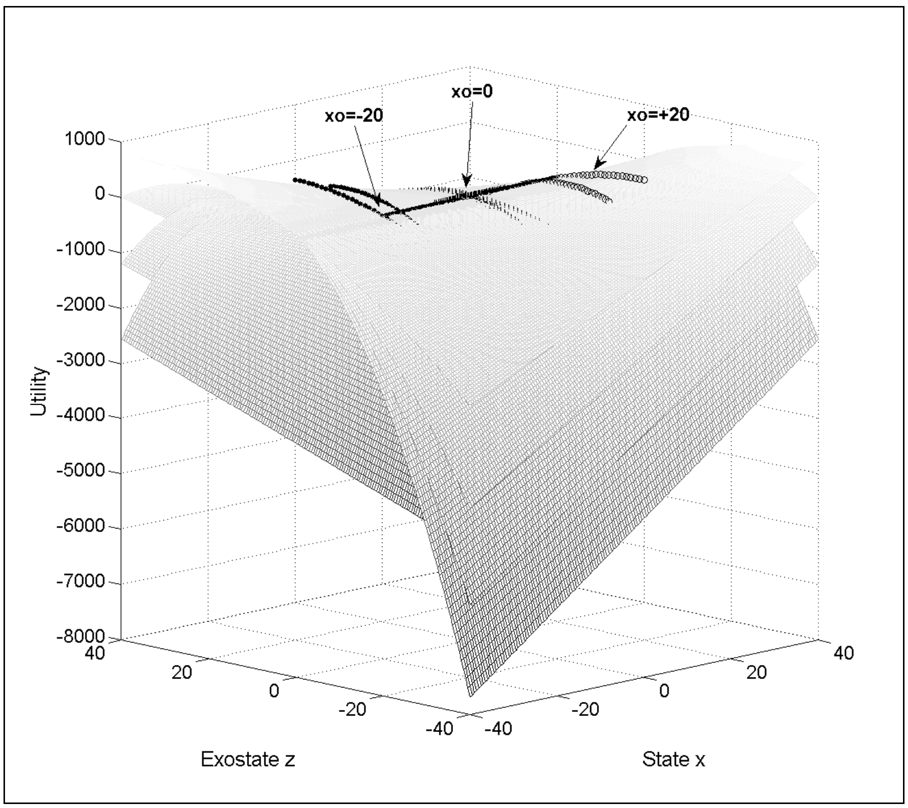

Figure 6.3. General and Specific (For Example #2) Utility Surfaces for the Constant-Step Exogenous-Input of (6.11) with $-20 \le C_1 \le 20$ and Selected Fixed-Values of x_0

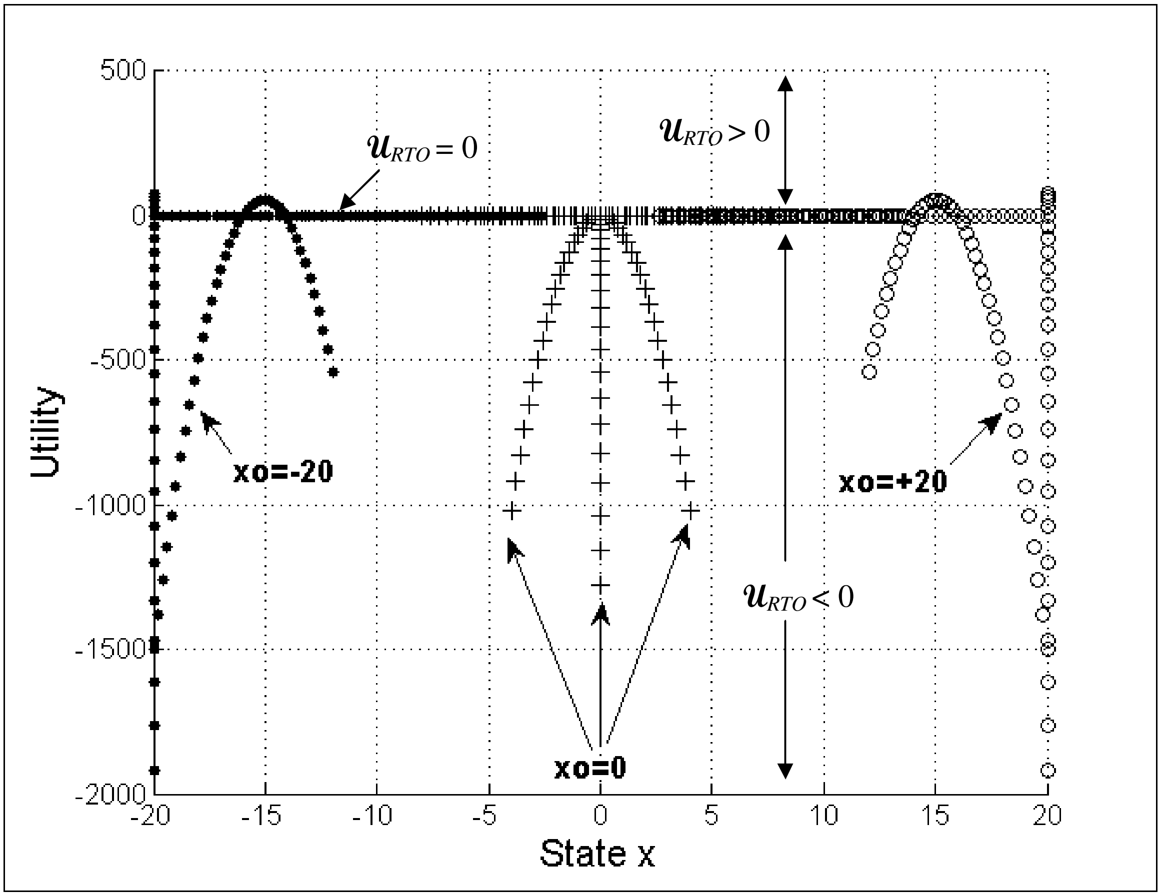

Figure 6.4. Rotated x-View of Positive and Negative Utility for the Constant-Step Exogenous-Input of (6.11) for Example #2; $-20 \leq C_1 \leq 20$ and $C_2=0$ and Selected Fixed-Values of x_0

From Figure 6.5, it can be seen that *positive* utility occurs (as a result of "EIU-smart" sequential-decisions) when C_1 takes the values of (0, +12) for the initial condition $x_0 = -20$ and when C_1 takes the values (-12, 0) for the initial condition $x_0 = +20$. Notice that the values of the constant-step exogenous-input (and in particular the values of C_1) that give a positive utility correspond to the same values obtained from the Optimization Criterion figure (Figure 6.2) when $x_0 = \pm 20$.

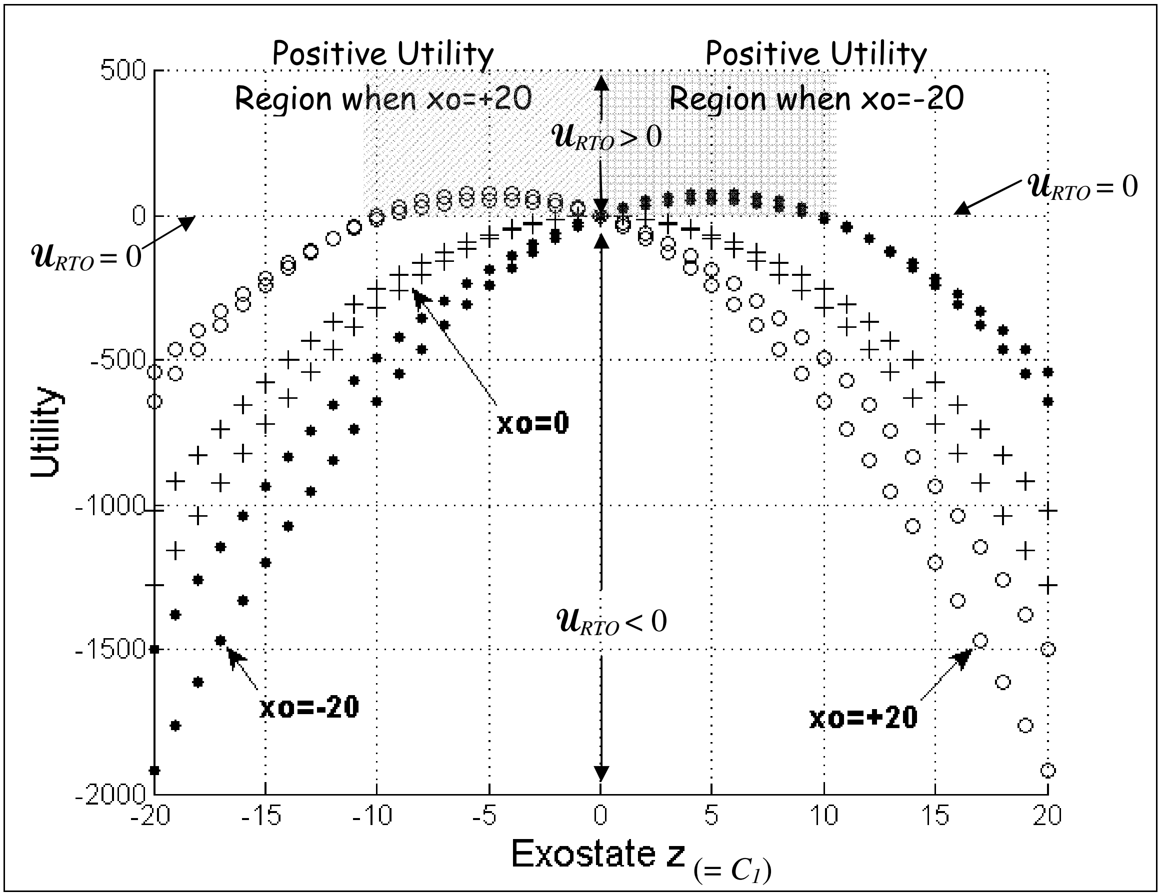

Figure 6.5. Rotated z-View of Utility of the Constant-Step Exogenous-Input of (6.11) for Example #2; $-20 \le C_1 \le 20$ and $C_2=0$ and Selected Fixed-Values of x_0

The zero-utility planes separating the domains of positive and negative utility in the (x, z)-space are defined by setting (6.26)-(6.29) equal to zero and solving for z_k as follows:

$$\mathcal{U}_0 = -\frac{8z_0}{5}\left(x_0 + 2z_0\right) = 0 \Rightarrow z_0 = 0, z_0 = -\frac{x_0}{2}, \tag{6.30}$$

$$\mathscr{U}_1 = -\frac{3z_1}{4}(2x_1 + 3z_1) = 0 \Rightarrow z_1 = 0, z_1 = -\frac{2x_1}{3}, \tag{6.31}$$

$$\mathscr{U}_2 = -\frac{4z_2}{3}(x_2 + z_2) = 0 \Rightarrow z_2 = 0, z_2 = -x_2, \tag{6.32}$$

$$\mathscr{U}_3 = -\frac{z_3}{2}(2x_3 + z_3) = 0 \Rightarrow z_3 = 0, z_3 = -2x_3. \tag{6.33}$$

Thus, at each decision stage, the positive utility domain in the *(x, z)*-space lies between the plane z_k =0 and the intersecting plane defined by (6.30)-(6.33) which can be generalized as described in Appendix E to give

$$z_k = 0; \quad z_k = 2Kx_k; \;\; K = -\frac{a^{N-k}}{\sum_{j=k}^{N-1} a^j f}, \tag{6.34}$$

where the K is a constant that varies with each decision stage k and corresponds to $[Kxz]^T[Kz]^{-1}$ in (5.22), a is the sequential process scalar coefficient in (6.1), f is the exogenous-input coefficient in (6.1), and N is the number of stages for the entire sequential-decision process ($N = T = 4$). The utility domains of (6.34) are shown in Figure 6.6.

The technical-effectiveness of the smart "exogenous-input utilizing" RTO-decision is quantified by the use of the expression in (5.24). Values for the effectiveness "$\mathcal{E}$" were computed for the current example in which the exogenous-input is a constant-step and plotted against the values of z. Figure 6.7 illustrates this effectiveness which clearly shows that the values of z that *further minimized* the otherwise "minimum-value" of the optimization criterion J in Figure 6.2, correspond to the same values of z in

Figure 6.7 for which the effectiveness "$\mathcal{E}$" is positive ($-12 \le z \le 12$). The effectiveness for this example reaches 60% at its highest value. This means that the "RTO-optimal" decision is skillfully "utilizing" the "disturbing-effects" caused-by the structured-variation exogenous-input of (6.11) to further minimize J by 60% *more than* would be possible in the case of no disturbance-input! This clearly demonstrates the wisdom of the admonition: "Never underestimate the possible utility presented by an unexpected-disturbance in a sequential-decision process" [39].

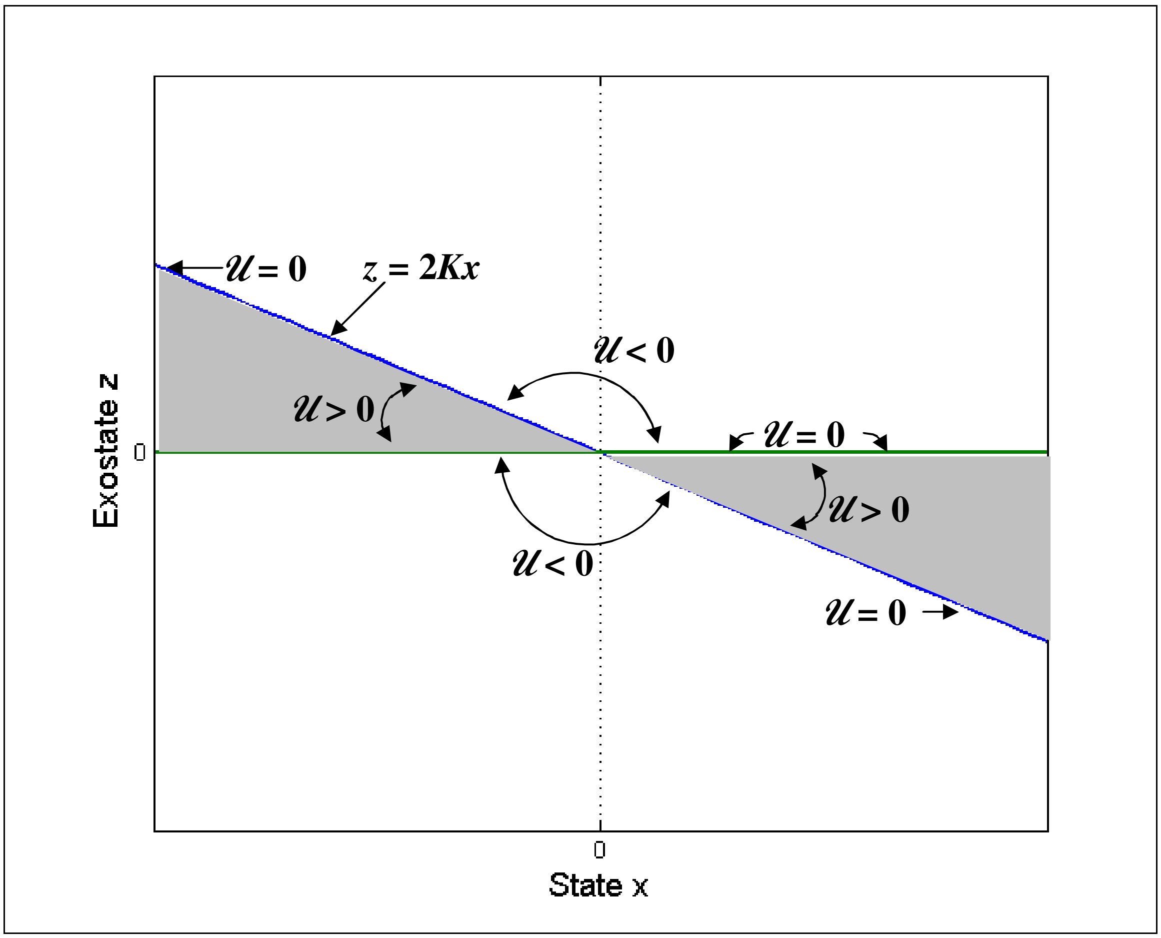

Figure 6.6. Positive, Negative, and Zero-Utility Domains in (x, z)-space for the Constant-Step in Example #2

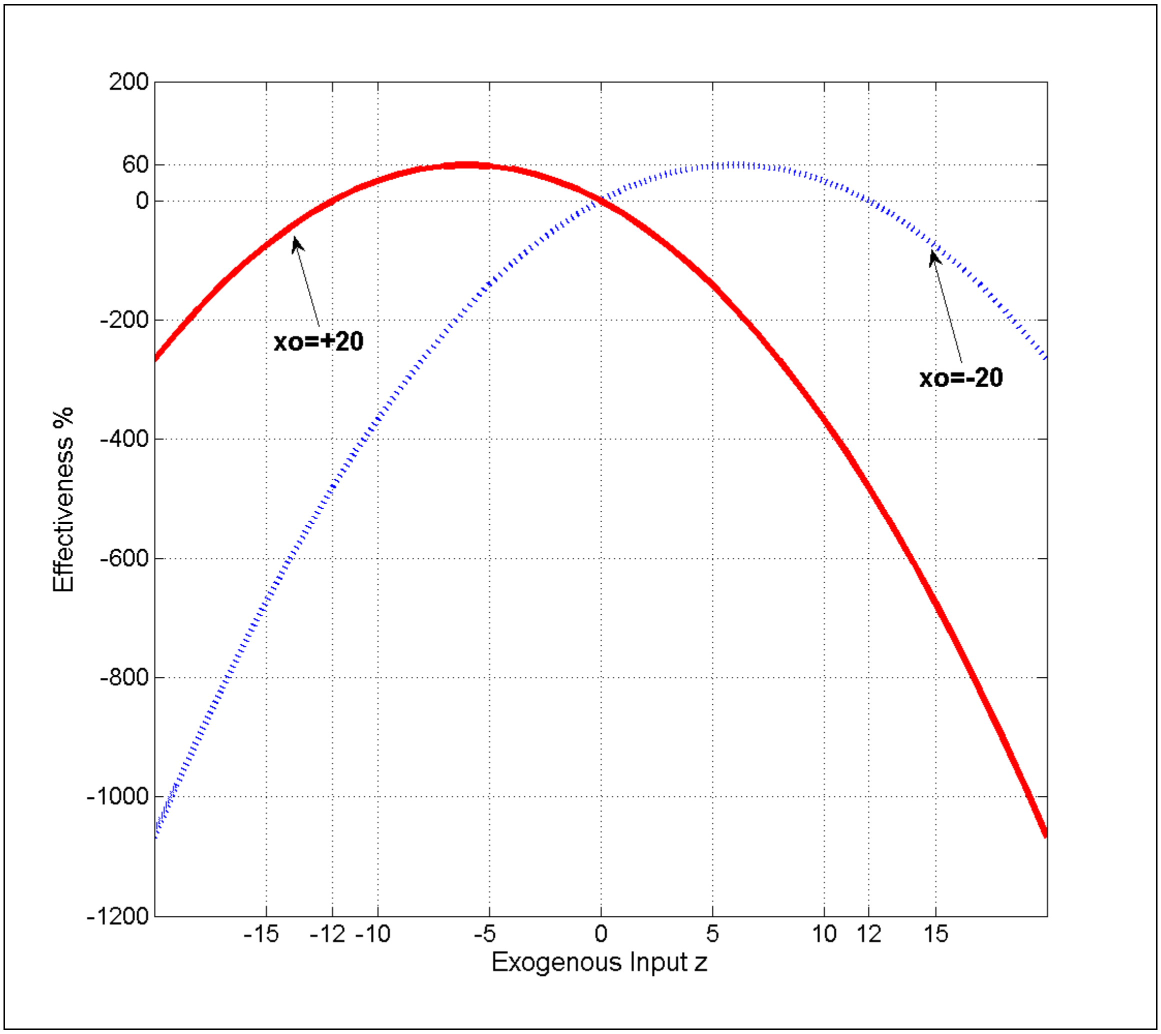

Figure 6.7. Effectiveness for Example #2

6.3 Example #3: The Case of an Unknown "Step Plus Ramp"-Type Exogenous-Input

For this example, the uncertain exogenous-input w_k is assumed to be approximated as an uncertain step plus a ramp. Thus, in discrete-time, the uncertain exogenous-input has the following "spline"-type form

$$w(t_i) = w_k = \begin{cases} B_1 + B_2 \cdot k & 0 \le k < t_2 \quad B_{1,2} = \text{unknown constants} \\ B_1 - B_2 \cdot k & t_2 \le k \le t_4 \quad t_2 = \text{ unknown} \end{cases} ; w_0 = B_1, \quad (6.35)$$

and is depicted in Figure 6.8 for B_1=1 and B_2=2.

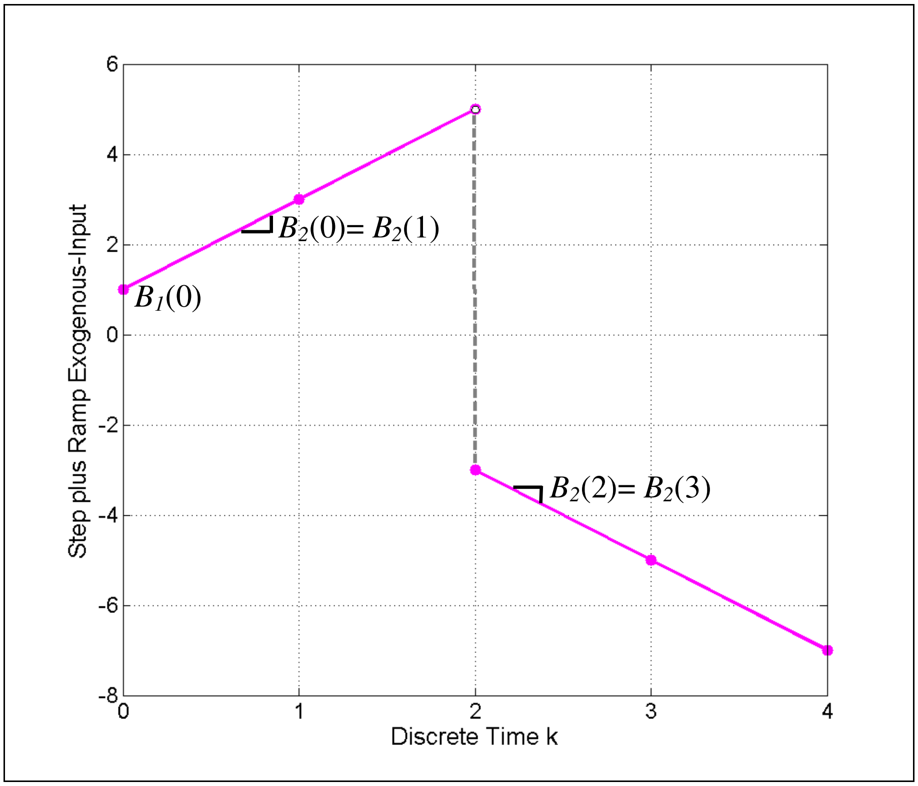

Figure 6.8. Step plus Ramp-Type Structured-Variation Exogenous-Input of Example #3 with B_1=1 and B_2=2

The discrete state-space model of the exogenous-input w_k is obtained by finding the proper (lowest) order difference equation for which (6.35) is the general solution as shown in Appendix D. Defining $z_1(k) \underline{\underline{\Delta}}\, w(k)$ and $z_2(k) \underline{\underline{\Delta}}\, w(k+1)$, the discrete state-space model for the exogenous-input (6.35) is as follows:

$$\begin{aligned}
&w(k) = [1 \quad 0]\begin{pmatrix} z_1(k) \\ z_2(k) \end{pmatrix}; \;\; H = [1 \quad 0] \\
&\begin{pmatrix} z_1(k+1) \\ z_2(k+1) \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}\begin{pmatrix} z_1(k) \\ z_2(k) \end{pmatrix} + \begin{pmatrix} \bar{\sigma}_1(k) \\ \bar{\sigma}_2(k) \end{pmatrix}; \;\; G = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix},
\end{aligned} \tag{6.36}$$

where the notation $z_1(k)$ denotes the value of the exogenous-input variable z_1 exostate at the decision-time k and $z_2(k)$ denotes the value of the exogenous-input exostate variable z_2 at the decision-time k. Similarly, $z_1(k+1)$ denotes the value of the exogenous-input exostate variable z_1 at the decision-time (k+1) and $z_2(k+1)$ indicates the value of the exogenous-input exostate variable z_2 at the decision-time (k+1).

The *composite state* $\tilde{x}$ consisting of the underlying system (process) state x and the "exostate z" of the structured-variation type exogenous-input is governed by the composite discrete time state-evolution model

$$\tilde{x} = \begin{bmatrix} x(k+1) \\ \hline z_1(k+1) \\ z_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 \\ \hline 0 & 0 & 1 \\ 0 & -1 & 2 \end{bmatrix} \cdot \begin{bmatrix} x(k) \\ \hline z_1(k) \\ z_2(k) \end{bmatrix} + \begin{bmatrix} 1 \\ \hline 0 \\ 0 \end{bmatrix} \cdot D_k + \begin{bmatrix} 0 \\ \hline \bar{\sigma}_1(k) \\ \bar{\sigma}_2(k) \end{bmatrix}, \;\; \tilde{x} \underline{\underline{\Delta}} \begin{pmatrix} x \\ \hline z_1 \\ z_2 \end{pmatrix}, \tag{6.37}$$

where k=0, 1, 2, 3, and N =T=4.

As in the solution-process used in the previous example, we first apply the Dynamic Programming technique in backward-time assuming that the $\bar{\sigma}_k \equiv 0$ in (6.37) on the time interval (3, 0). This is equivalent to the exogenous-input of (6.35) maintaining a constant slope throughout the entire backward-recursive process, (3, 0). This step will give us the set of optimal decisions as defined by (4.14) that can then be applied in forward-time using the RTO principle. As stated in the preceding example, to

apply the RTO principle in forward time, we will assume an arbitrary, known initial exogenous-input value, w_0. In keeping with the RTO Principle, we will assume that the values of B_1 and B_2 in the exogenous-input will remain unchanged from the current time, k=0, until the end of the process at k=4. However, when the structured-variation exogenous-input constants B_1 and B_2 change *unexpectedly* to a different value at t_2, we will use the backward time optimal decision corresponding to the *new value* of the structured-variation exogenous-input which will be updated with the composite state information via the implied real-time composite-state observer. In this way, the effect of any post-decision "arrivals" of $\overline{\sigma}_k$ will be accounted for *at the very-next decision time* in the Dynamic Programming/RTO solution process.

Thus, assuming that the $\overline{\sigma}_k \equiv 0$ in (6.37), the backward-time sequence of scalar RTO-optimal decisions $D_k^*(x_k, z_k)$ with k=3, 2, 1, 0 can be easily computed via (4.14) with a slight variation to accommodate the second-order exogenous-input exostate $col.(z_1 \mid z_2)$ of this example as follows:

$$D_k^* = -\left[\frac{a^{(N-1-k)}b}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)}b^2}\right] \times \left[a^{(N-k)}x_k + \sum_{j=k}^{N-1}\left(a^{(N-1-j)}\right) fH\overrightarrow{z_j}\right], \tag{6.38}$$

where the matrix H is a 1x2 matrix and $\overrightarrow{z_j}$ is a 2x1 vector. The product $H\overrightarrow{z_j}$ is a scalar and due to the nature of H ($H = \begin{bmatrix}1 & 0\end{bmatrix}$), the RTO-optimal decisions have the following scalar form which only involves the first exogenous-input exostate $z_1(k)$

$$D_3^* = -\frac{(x_3 + z_1(3))}{2} \tag{6.39}$$

$$D_2^* = -\frac{(x_2 + z_1(2) + z_1(3))}{3} \tag{6.40}$$

$$D_1^* = -\frac{(x_1 + z_1(1) + z_1(2) + z_1(3))}{4} \tag{6.41}$$

$$D_0^* = -\frac{(x_0 + z_1(0) + z_1(1) + z_1(2) + z_1(3))}{5}, \tag{6.42}$$

where the notation $z_1(k)$ indicates the first exogenous-input state z_1 at time k.

Next, in forward-time and starting at t_0=0, we apply the previously calculated backward-time optimal decisions (6.39)-(6.42) which were determined by normal Dynamic Programming calculations. Notice that at time k=0, 1, and 2, the equation for the optimal decision D_k is defined in terms of the value of the exogenous-input at future times which are unknown to the decision maker. However, according to the state-space model (6.36), the values of the exogenous-input at k=0 are defined as

$$\begin{aligned}
&z_1(0) = w(0) = B_1; \\
&z_2(0) = w(1) = B_1 + B_2; \\
&z_1(1) = z_2(0) = B_1 + B_2; \\
&z_2(1) = -z_1(0) + 2z_2(0) = B_1 + 2B_2; \\
&z_1(2) = z_2(1) = B_1 + 2B_2; \\
&z_2(2) = -z_1(1) + 2z_2(1) = B_1 + 3B_2; \\
&z_1(3) = z_2(2) = B_1 + 3B_2; \\
&z_2(3) = -z_1(2) + 2z_2(2) = B_1 + 4B_2,
\end{aligned} \tag{6.43}$$

where the initial value of the "exostate" $z_i(0)$ is assumed known and we have applied the RTO Principle, thus assuming that the $\sigma_i = 0$ (B_1, B_2 will *remain constant*) from the current time *until the end* of the process; however, that assumption is re-evaluated at each time step and the actual current-values of the (B_1, B_2) are factored-into the RTO-decision at the very next decision-time. Thus, at time k=0, with $\sigma_i(0) = \sigma_i(1) = \sigma_i(2) = 0$, the decision D_0 in (6.42) becomes

$$D_0 = -\frac{(x_0 + 4B_1 + 6B_2)}{5}. \qquad (6.44)$$

At time k=1 and according to (6.36), the values of the exogenous-input $z_i(1)$ are determined to be

$$\begin{aligned}
&z_1(1) = w(1) = B_1 + B_2;\\
&z_2(1) = w(2) = B_1 + 2B_2;\\
&z_1(2) = z_2(1) = B_1 + 2B_2;\\
&z_2(2) = -z_1(1) + 2z_2(1) = B_1 + 3B_2;\\
&z_1(3) = z_2(2) = B_1 + 3B_2;\\
&z_2(3) = -z_1(2) + 2z_2(2) = B_1 + 4B_2,
\end{aligned} \qquad (6.45)$$

where the value of the exogenous-input $z_i(1)$ was determined by the implied composite-state observer and the RTO principle was applied, therefore maintaining the one decision-to next decision assumption that the constants B_1 and B_2 will remain unchanged until the end of the process ($\sigma_i = 0$), and thus, at time k=1, $\sigma_i(1) = \sigma_i(2) = 0$ and the RTO decision D_1 in (6.41) becomes

$$D_1 = -\frac{(x_1 + 3B_1 + 6B_2)}{4}. \qquad (6.46)$$

At time k=2 and according to (6.36), the value of the exogenous-input $z_i(2)$ is determined to be

$$\begin{aligned}
&z_1(2) = w(2) = B_1 - 2B_2; \\
&z_2(2) = w(3) = B_1 - 3B_2; \\
&z_1(3) = z_2(2) = B_1 - 3B_2; \\
&z_2(3) = -z_1(2) + z_2(2) = -B_1 + 2B_2 + B_1 - 3B_2 = -B_2.
\end{aligned} \tag{6.47}$$

Notice that at k=2, the change in the value of the exogenous-input was estimated by the implied composite-state observer and thus, it embodies all the information needed to make a rational, non-gambling *real-time* decision at time k=2. Once again, we have assumed (until the next decision-time) that the value of the constants B_1 and B_2 of the exogenous-input will remain at this new value until the *end of the process* and thus, $\sigma_i(2) = 0$ and the RTO decision D_2 in (6.40) becomes

$$D_2 = -\frac{\left(x_2 + 2B_1 - 5B_2\right)}{3}. \tag{6.48}$$

At k=3, according to (6.36), $z_1(3) = B_1 - 3B_2$ and $z_2(3) = B_1 - 4B_2$ and the RTO decision D_3 in (6.39) becomes

$$D_3 = -\frac{\left(x_3 + B_1 - 3B_2\right)}{2}. \tag{6.49}$$

Thus, at the end of the process at k=T=4, the RTO "value" of the optimization criterion J becomes

$$J = (x_T)^2 + \left[-\frac{(x_0 + 4B_1 + 6B_2)}{5} \right]^2 + \left[-\frac{(x_1 + 3B_1 + 6B_2)}{4} \right]^2 + \left[-\frac{(x_2 + 2B_1 - 5B_2)}{3} \right]^2 + \left[-\frac{(x_3 + B_1 - 3B_2)}{2} \right]^2, \qquad (6.50)$$

where x_T is the terminal state obtained at the end of the sequential-decision process.

Figure 6.9 shows the value of the RTO-optimization criterion for the step plus ramp of (6.35) with three different initial state conditions x_0. The lines in Figure 6.9 were obtained by varying B_1 with $-20 \le B_1 \le 20$ and letting B_2=2. The plus (+) line represents J for the case in which x_0=0, the circle (o) line is for the case x_0=+20, and the dotted line is the case x_0=-20. Figure 6.9 also shows the value for the optimization criterion of Example #1. Thus, the gray dashed line represents the value of J for the baseline case with a zero exogenous-input as described in (6.10) for the case of x_0=±20.

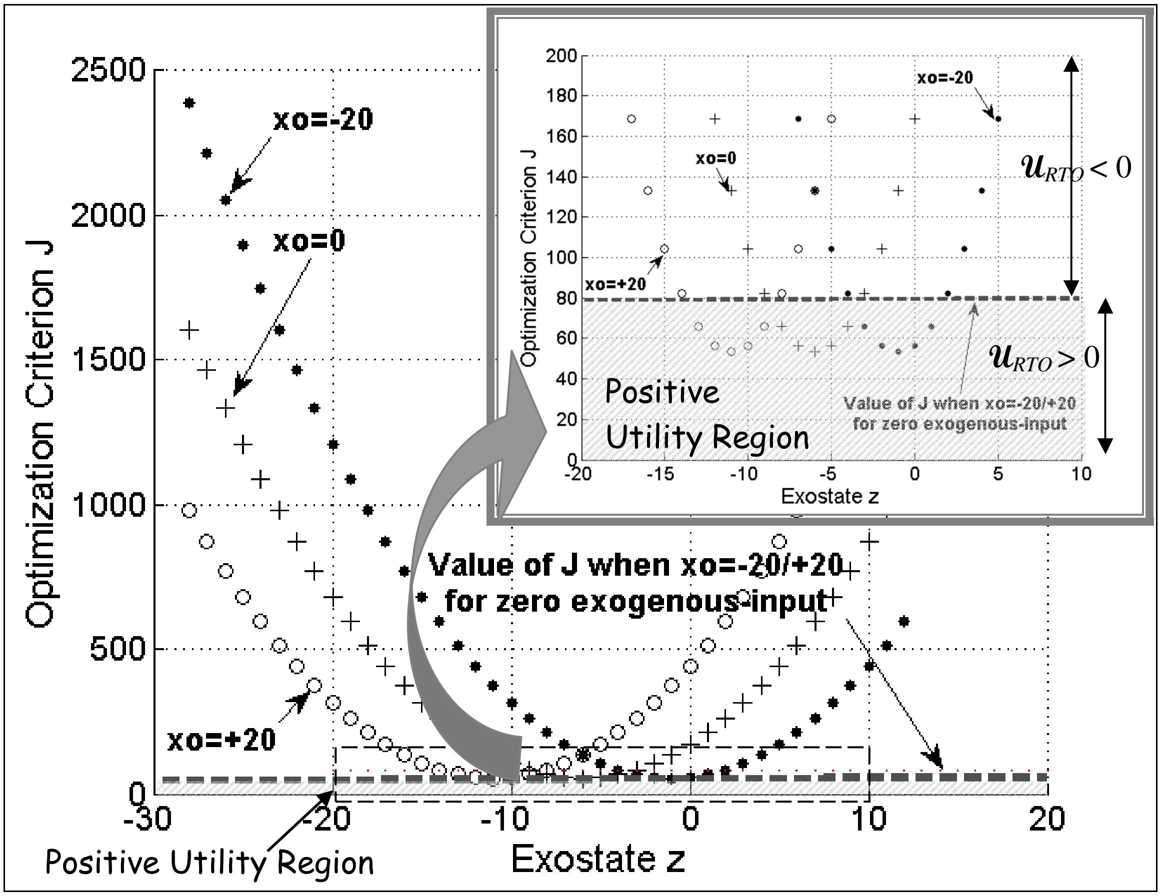

Figure 6.9. Exogenous-Input Utility and Optimal Values *J* of the Criteria of Optimality for Step plus Ramp and Zero Exogenous-Inputs with Area of Interest Zoomed-In; calculated from (6.50)

Figure 6.9 shows the positive "utility" of the step plus ramp exogenous input when compared to the baseline case (zero exogenous-input represented by the dashed line). By comparing the value of the optimization criterion *J* when there is no exogenous-input present (i.e., $w_k \equiv 0$ and *J=80*) to the value of *J* when the exogenous-input is present and is a step plus ramp with x_0=±20, one can observe that the "optimum" value of *J* is further minimized when the value of the step plus ramp exogenous-input falls in the interval (-4,2) for x_0=-20 and when *z* belongs to the open interval (-14, -8) for

the case x_0=+20. The minimum possible value of J when the exogenous-input is not present and x_0=±20 is *80*. On the other hand, when the exogenous-input is a step plus ramp as described by (6.35), the value of J *is lowered* to less than *80* when z takes certain values. This area of positive utility is depicted by the shaded region in Figure 6.9.

The value of the instantaneous utility is derived from (5.20) which, due to the second-order exogenous-input exostate $col.(z_1 \mid z_2)$, has the slightly modified form

$$\mathscr{U}_{RTO}(k) = -\left[\frac{1}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)} b^2}\right] \times \left[2a^{(N-k)} x(k) \sum_{j=k}^{N-1}\left(a^{(N-1-j)} fH\overrightarrow{z(j)}\right) + \left(\sum_{j=k}^{N-1}\left(a^{(N-1-j)} fH\overrightarrow{z(j)}\right)\right)^2\right], \tag{6.51}$$

where H is the 1x2 matrix defined in (6.36) and $\overrightarrow{z(j)}$ represents the 2x1 vector which describes the step plus ramp exogenous-input of this example. According to (6.51), the value of the instantaneous utility is still a scalar at each discrete-time and is as follows

$$\begin{aligned}\mathscr{U}_0 &= -\frac{(z_1(0)+z_1(1)+z_1(2)+z_1(3))(z_1(0)+z_1(1)+z_1(2)+z_1(3)+2x_0)}{5} \\ &= -\frac{(4B_1+6B_2)(4B_1+6B_2+2x_0)}{5}\end{aligned} \tag{6.52}$$

$$\begin{aligned}\mathscr{U}_1 &= -\frac{(z_1(1)+z_1(2)+z_1(3))(z_1(1)+z_1(2)+z_1(3)+2x_1)}{4} \\ &= -\frac{(3B_1+6B_2)(3B_1+6B_2+2x_1)}{4}\end{aligned} \tag{6.53}$$

$$\mathscr{U}_2 = -\frac{(z_1(2)+z_1(3))(z_1(2)+z_1(3)+2x_2)}{3} = -\frac{(2B_1-5B_2)(2B_1-5B_2+2x_2)}{3} \tag{6.54}$$

$$\mathcal{U}_3 = -\frac{z_1(3)(z_1(3)+2x_3)}{2} = -\frac{(B_1-3B_2)(B_1-3B_2+2x_3)}{2}. \tag{6.55}$$

Figure 6.10 shows the values of the utility in the ($\boldsymbol{\mathcal{U}}$-x)-space for the specific Example #3 in which the exogenous-input z is a step plus ramp that takes the values described by (6.35) and the state x only takes the values given by (6.1). The lines in Figure (6.10) were obtained by determining the utility for the entire process k =0, 1, 2, 3 with the step plus ramp exogenous-input as described by (6.35) for a specific value of B_1 and B_2=2. Then a new value was selected for B_1 while B_2 was held at a value of 2 and the utility at each decision time computed again. This process was repeated for values of the step plus ramp exogenous input in which $-20 \leq B_1 \leq 20$ and B_2=2. The plus (+) line represents the value of the instantaneous utility for an initial state x_0=0, while the circle (o) line corresponds to x_0=+20 and the dotted line corresponds to x_0=-20.

In Figure 6.10 one can observe that the optimal state x_k progressively moves through regions of positive utility $\boldsymbol{\mathcal{U}}_{RTO}>0$ and negative utility $\boldsymbol{\mathcal{U}}_{RTO}<0$. This demonstrates that the step plus ramp exogenous-input contributes to *further minimizing* the optimization criterion J in certain regions of the ($\boldsymbol{\mathcal{U}}$-x-z)-space. This is a clear indication that certain values of the step plus ramp exogenous-input of (6.35) acting on the state described by (6.1) and with and optimization criterion as described by (6.2) are in fact "useful" in further decreasing the value of the optimization criterion J provided that the decisions are EIU-"smart" to enable "harvesting" of the utility embodied in the exogenous-input w_k.

Figure 6.11 shows the utility for this example in the ($\mathcal{U}$-z)-space. The shaded areas in Figure 6.11 represent the overlapping regions of positive utility for x_0=±20. Notice that these areas are for the instantaneous utility at each discrete time k and are larger than those represented in the Optimization Criterion figure (Figure 6.9) since the optimization criterion J is calculated at the end of the sequential decision process and the utility is calculated at each discrete-time k.

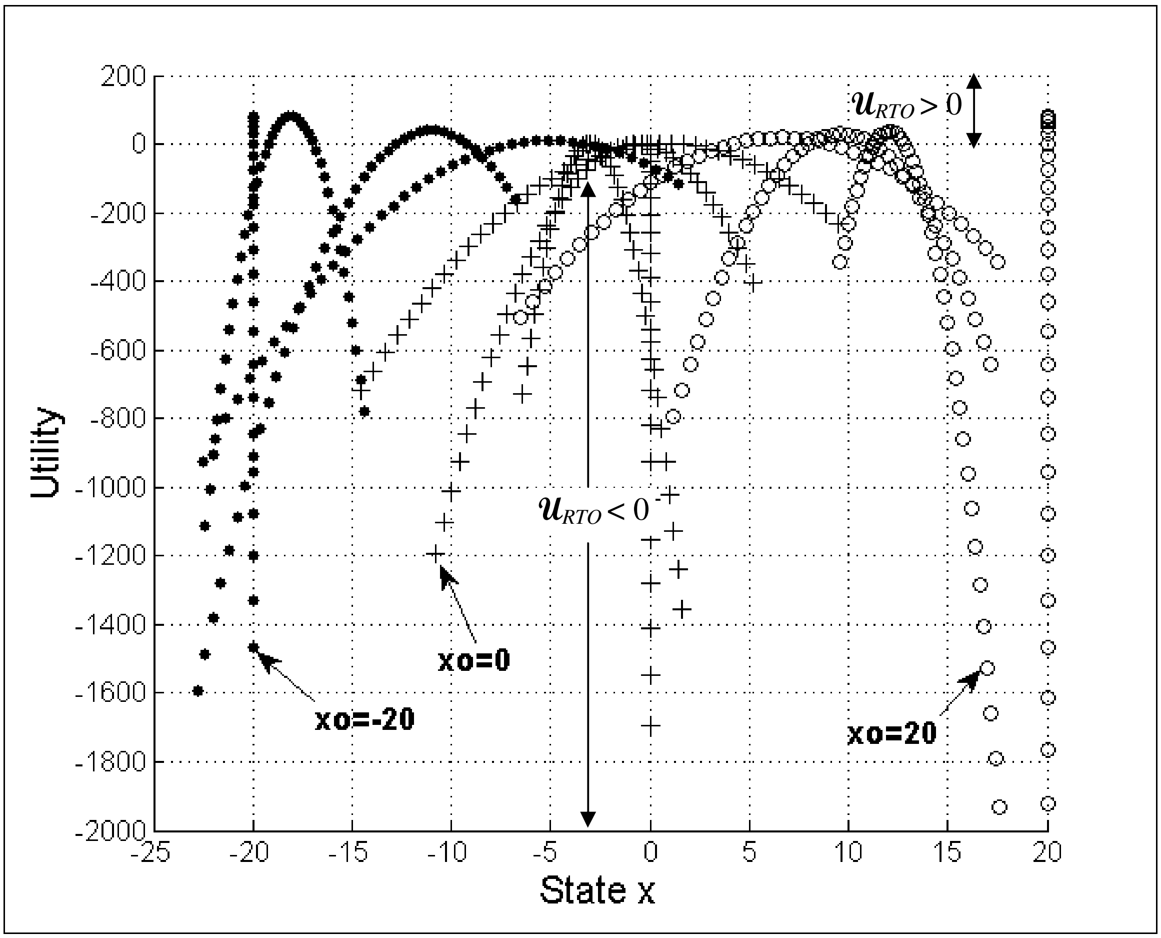

Figure 6.10. x-Axis Projection (View) of Utility-Values for a Step plus Ramp-Type Exogenous-Input in Example #3; -20≤B_1≤20 and B_2=2 and Selected Fixed-Values of x_0

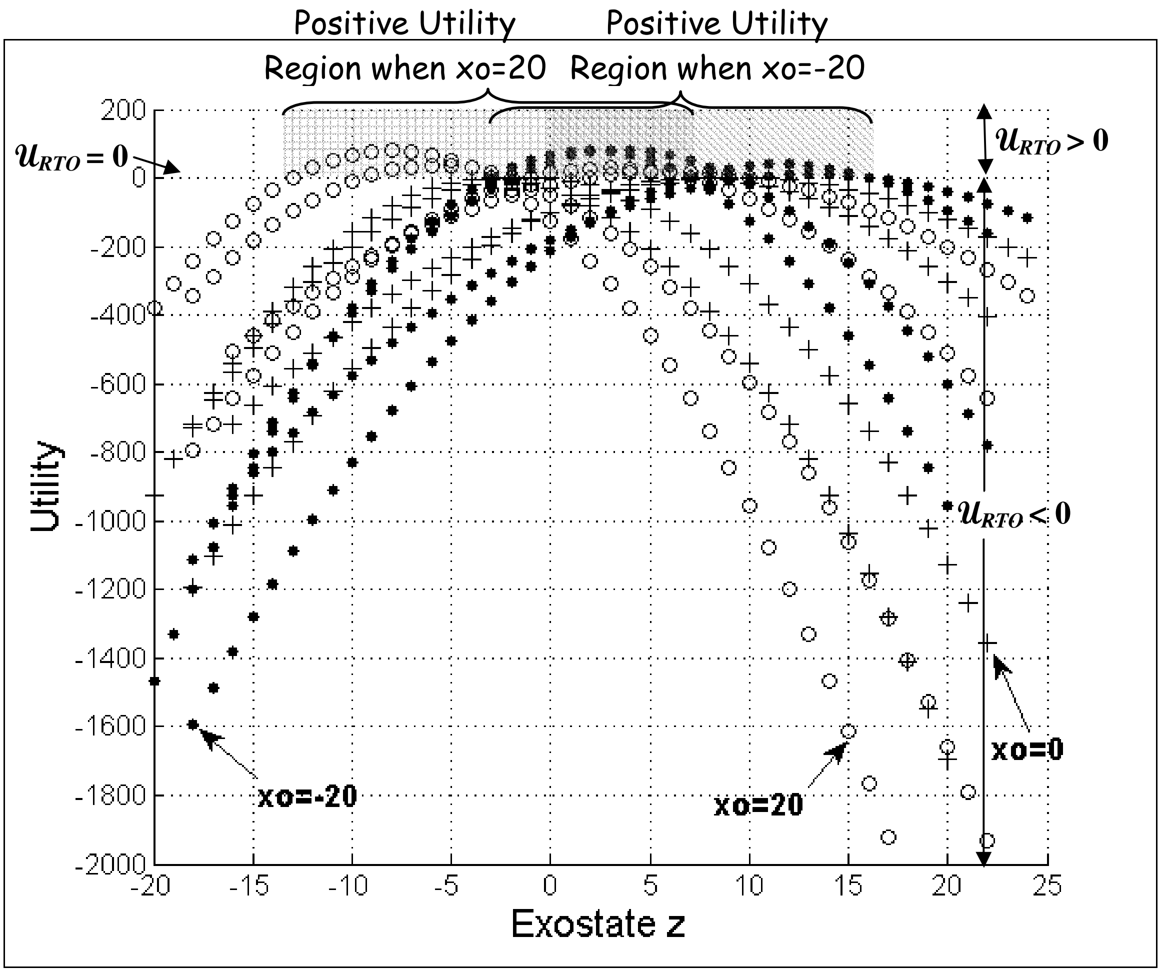

Figure 6.11. *z*- Axis Projection (View) of the Utility-Value for the Step plus Ramp Exogenous-Input in Example #3; $-20 \le B_1 \le 20$ and $B_2=2$ and Selected Fixed-Values of x_0

The domains of utility in the (x, z)-plane for this example at each discrete-time k depend on the value of the exogenous input state-variable $z_1(k)$ and the process state $x(k)$ and can be deduced by setting the utility expression (6.51) equal to zero and solving for z_k. For our specific example, we obtain the following specific utility domains at each discrete-time k

$$z_1(0) = 0 \text{ and } z_1(0) = -x(0)/2 \tag{6.56}$$

$$z_1(1) = 0 \text{ and } z_1(1) = -2x(1)/3 \tag{6.57}$$

$$z_1(2) = 0 \text{ and } z_1(2) = -x(2) \tag{6.58}$$

$$z_1(3) = 0 \text{ and } z_1(3) = -2x(3)\,, \tag{6.59}$$

which can be generalized as shown in Appendix E to be

$$z_1(k) = 0 \text{ and } z_1(k) = 2Kx(k);\ K = -\frac{a^{(N-k)}}{\sum_{j=k}^{N-1}(a^j f)}, \tag{6.60}$$

and where K represents a scalar quantity that changes at each discrete-time k, a is the sequential process scalar coefficient in (6.1), f is the exogenous-input coefficient in (6.1), and N is the number of stages for the entire sequential-decision process (N =T= 4). Figure 6.12 shows the general utility domain (6.60) for the step plus ramp exogenous-input of this Example. Notice that the utility domain for this example depends exclusively on the value of the exogenous input state-variable $z_1(k)$ and not on the value of the exogenous-input variable $z_2(k)$ due to the nature of the matrix H (H=[1 0]). This can be observed in the summation term of the utility expression (6.51) which has been extracted below

$$\sum_{j=k}^{N-1}(a^{(N-1-j)} fH\overline{z(j)}) \Rightarrow fH\overline{z(j)} = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}z_1(k)\\ z_2(k)\end{bmatrix} = z_1(k)\,, \tag{6.61}$$

where f is the exogenous-input coefficient which is equal to 1 for our example, and a is the sequential-decision scalar coefficient also equal to 1 for this example. Also notice that the utility domains for this example in which the exogenous-input is a Step plus Ramp are the same as the utility domains for the Stepwise constant.

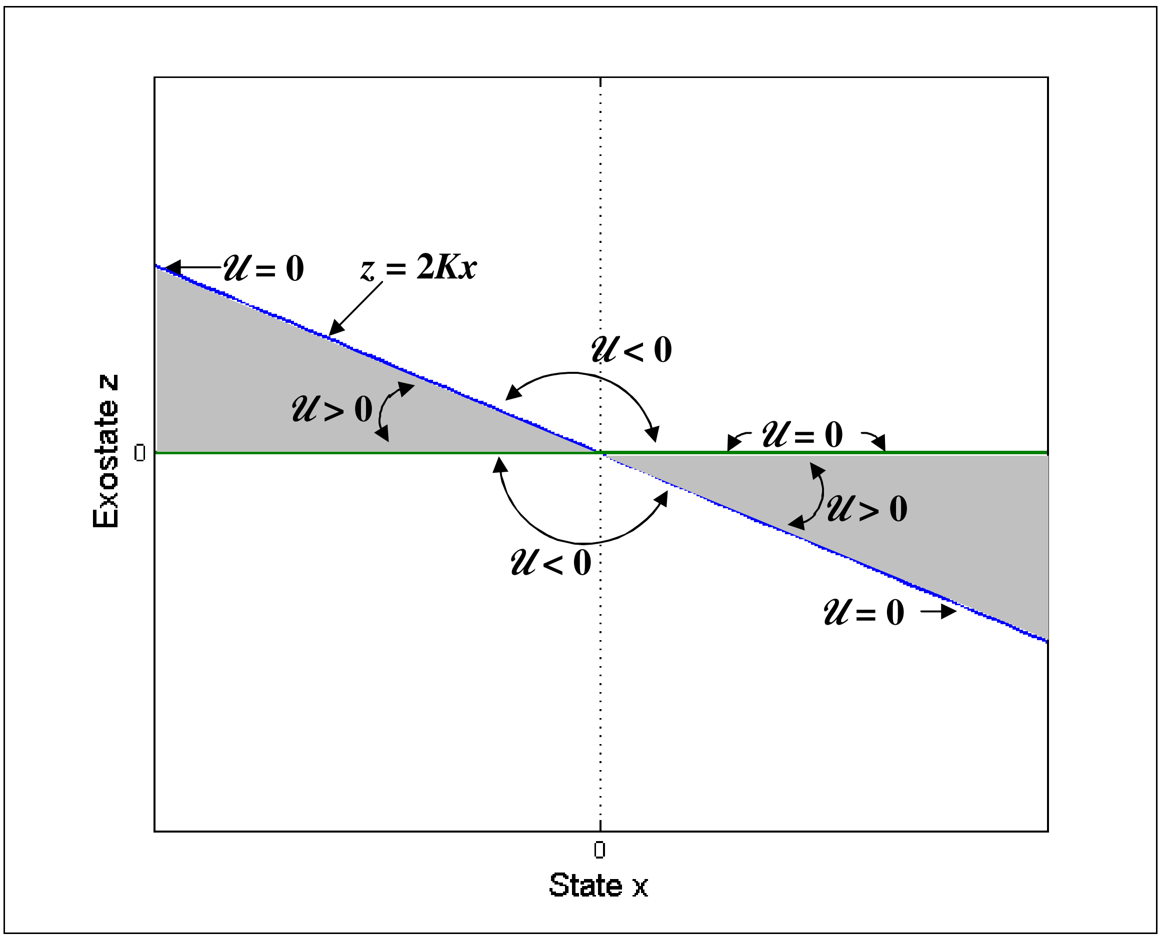

Figure 6.12. Illustration of Generic Utility Domains for the Step plus Ramp Exogenous-Input in Example #3

The effectiveness of the step plus ramp "exogenous-input-utilizing (EIU-smart) decision" is based on expression (5.24) and is shown in Figure 6.13. Values for the effectiveness $\mathcal{E}$ were computed for the current example in which the exogenous-input is a step plus ramp and plotted against the values of z. The values of z which give a positive effectiveness $\mathcal{E}>0$ correspond to the <u>same values</u> of z that further minimized the optimization criterion J in Figure 6.9, that is, for x_0=-20, positive effectiveness occurs in the interval (-4, 2) and for x_0=+20, positive effectiveness occurs in the interval (-14, -8). The highest value of effectiveness $\mathcal{E}$ for this example is 33.3% which occurs when x_0=-20 and z=-1. For the case when x_0=+20, the maximum effectiveness is also 33.3% which occurs when z=-11.

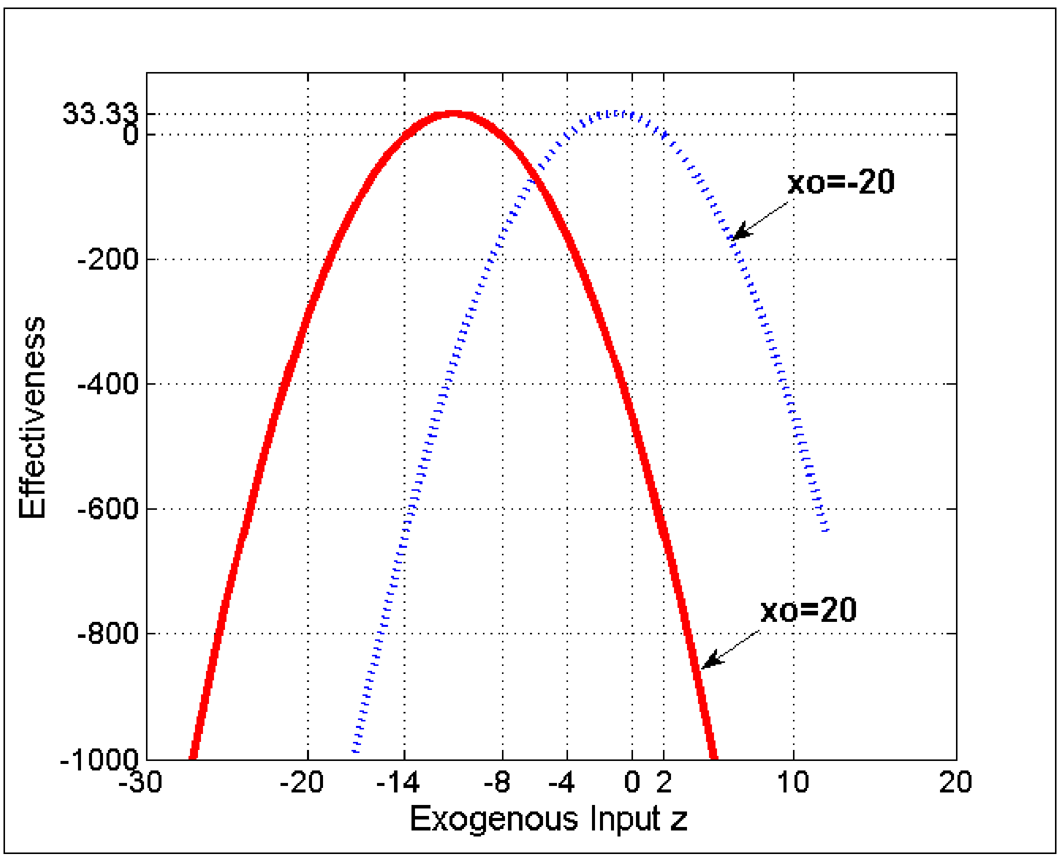

Figure 6.13. Plots of the Effectiveness for Example #3 for x_o=±20; -20≤B_1≤20 and B_2=2

6.4 Example #4: A More-General "Unknown Exogenous-Input"

For this example, the uncertain exogenous-input is chosen to represent a broad class of uncertain, meandering-type functions described by the following continuous-time "cubic polynomial spline" expression

$$w(k) = C_1 + C_2 k + C_3 k^2 + C_4 k^3;\ C_i = \text{unknown constant}\ . \tag{6.62}$$

Figure 6.14 depicts this cubic polynomial spline with C_1=3, C_2=-5, C_3=-1.3 and C_4=1.

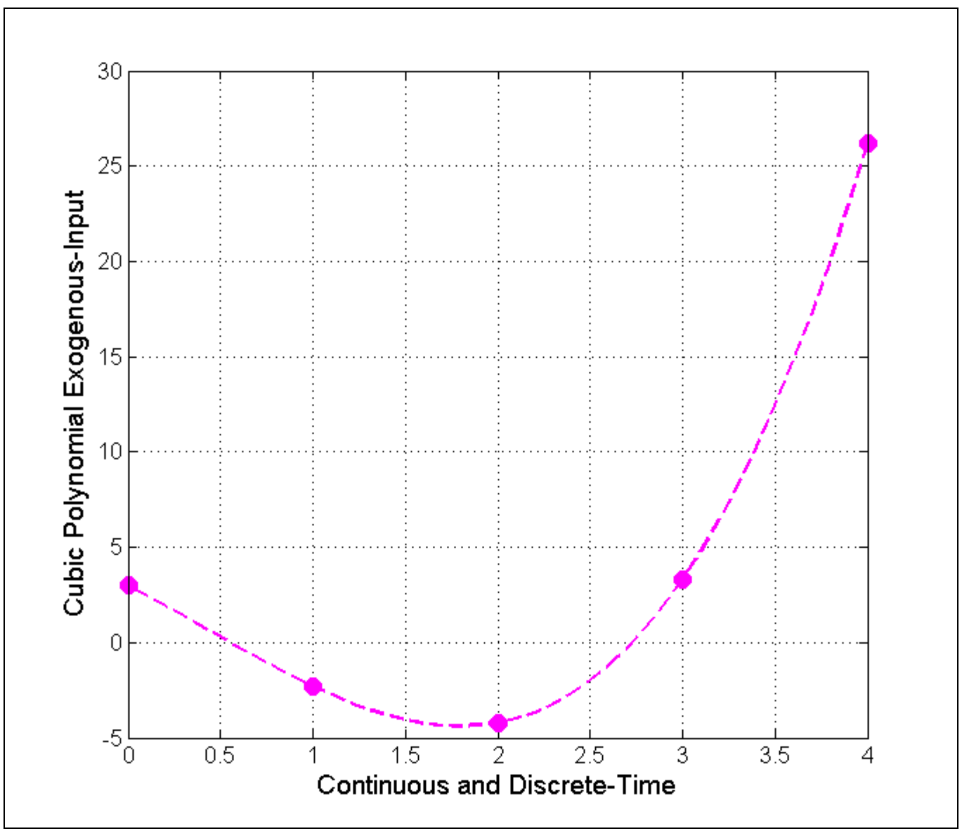

Figure 6.14. Cubic Polynomial-Spline Type Exogenous-Input for Example #4 with C_1=3, C_2=-5, C_3=-1.3 and C_4=1

To define the state-space model of the exogenous-input (6.62), we must find the proper (lowest) order difference-equation for which (6.62) is a solution and then obtain the exogenous-input state-space model [35] as shown in Appendix D. The discrete state-space model of the exogenous-input w_k is derived in Appendix D and is

$$w(k) = H \cdot z; H = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix};$$
$$\begin{bmatrix} z_1(k+1) \\ z_2(k+1) \\ z_3(k+1) \\ z_4(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & -6 & 4 & -1 \end{bmatrix} \begin{bmatrix} z_1(k) \\ z_2(k) \\ z_3(k) \\ z_4(k) \end{bmatrix} + \begin{bmatrix} \bar{\sigma}_1(k) \\ \bar{\sigma}_2(k) \\ \bar{\sigma}_3(k) \\ \bar{\sigma}_4(k) \end{bmatrix}; G = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & -6 & 4 & -1 \end{bmatrix}, \tag{6.63}$$

where the notation $z_1(k)$ denotes the value of the exogenous-input state-variable z_1 at decision-time *k*, $z_2(k)$ denotes the value of the exogenous-input state-variable z_2 at decision-time *k* and so on. Similarly, $z_1(k+1)$ denotes the value of the exogenous-input state-variable z_1 at time (*k*+1), $z_2(k+1)$ denotes the value of the exogenous-input state-variable z_2 at time (*k*+1) and likewise for the other exostate variables.

The *composite state* $\tilde{x}$ consisting of the underlying system (process) state *x* and the structured-variation type exogenous-input "exostate" *z* is governed by the discrete time state-evolution model

$$\tilde{x} = \left[\begin{array}{c} x(k+1) \\ \hline z_1(k+1) \\ z_2(k+1) \\ z_3(k+1) \\ z_4(k+1) \end{array}\right] = \left[\begin{array}{ccccc} 1 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 4 & -6 & 4 & -1 \end{array}\right] \left(\begin{array}{c} x(k) \\ \hline z_1(k) \\ z_2(k) \\ z_3(k) \\ z_4(k) \end{array}\right) + \left[\begin{array}{c} 1 \\ \hline 0 \\ 0 \\ 0 \\ 0 \end{array}\right] \cdot D_k + \left[\begin{array}{c} 0 \\ \hline \bar{\sigma}_1(k) \\ \bar{\sigma}_2(k) \\ \bar{\sigma}_3(k) \\ \bar{\sigma}_4(k) \end{array}\right], \quad \tilde{x} \underline{\underline{\Delta}} \left(\begin{array}{c} x \\ \hline z_1 \\ z_2 \\ z_3 \\ z_4 \end{array}\right), \tag{6.64}$$

where *k*=0, 1, 2, 3, *N*=*T*=4.

We apply the RTO principle in forward time using the previously calculated backward-time sequence of scalar RTO-optimal decisions $D_k^*(x_k, z_k)$ with k=3, 2, 1, 0 as described by (6.38). We will assume an arbitrary *known* initial exogenous-input value w_0 and tacitly assume that the exogenous-input's constants C_i in (6.62) will remain unchanged from the current time, k=0, until the end of the process at k=4 ($\overline{\sigma}_k \equiv 0$ assumption in (6.63)). However, when the structured-variation exogenous-input changes *unexpectedly* to a different value, the composite-state observer will automatically update the new value of the structured-variation exogenous-input at the very-next decision-time. In this way the real-time decisions based-on the RTO Principle "optimally" account for the effect of unpredictable between-decision "arrivals" of the time-sparse $\overline{\sigma}_k$ in the Dynamic Programming process.

For this example, the matrix H is a 1x4 matrix, f is a scalar and $\overrightarrow{z_j}$ is a 4x1 vector. The product $fH\overrightarrow{z_j}$ is a scalar and due to the nature of H ($H = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}$), the RTO-optimal decisions depend only on the exostate element $z_1(k)$. Thus, the RTO-optimal decisions at each discrete-time k are scalars and have the following form

$$\begin{aligned} D_k^* &= -\left[\frac{a^{(N-1-k)}b}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)}b^2}\right] \times \left[a^{(N-k)}x(k) + \sum_{j=k}^{N-1}\left(a^{(N-1-j)} fH\overrightarrow{z(j)}\right)\right] \\ &= -\left[\frac{a^{(N-1-k)}b}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)}b^2}\right] \times \left[a^{(N-k)}x(k) + \sum_{j=k}^{N-1}\left(a^{(N-1-j)} z_1(j)\right)\right]. \end{aligned} \tag{6.65}$$

Thus, in <u>forward</u>-time and starting at k=0, the values of the cubic polynomial-spline type exogenous-input are defined according to (6.63) as

$$
\begin{aligned}
&z_1(0) = w_0 = C_1 + C_2k + C_3k^2 + C_4k^3 = C_1,\\
&z_2(0) = C_2 + 2C_3k + 3C_4k^2 = C_2,\\
&z_3(0) = 2C_3 + 6C_4k = 2C_3,\\
&z_4(0) = 6C_4,\\
&z_1(1) = z_2(0) = C_2,\\
&z_2(1) = z_3(0) = 2C_3,\\
&z_3(1) = z_4(0) = 6C_4,\\
&z_4(1) = 4z_1(0) - 6z_2(0) + 4z_3(0) - z_4(0) = 4C_1 - 6C_2 + 8C_3 - 6C_4,\\
&z_1(2) = z_2(1) = 2C_4,\\
&z_2(2) = z_3(1) = 6C_4,\\
&z_3(2) = z_4(1) = 4C_1 - 6C_2 + 8C_3 - 6C_4,\\
&z_4(2) = 4z_1(1) - 6z_2(1) + 4z_3(1) - z_4(1) = -4C_1 + 10C_2 - 20C_3 + 30C_4,\\
&z_1(3) = z_2(2) = 6C_4,\\
&z_2(3) = z_3(2) = 4C_1 - 6C_2 + 8C_3 - 6C_4,\\
&z_3(3) = z_4(2) = -4C_1 + 10C_2 - 20C_3 + 30C_4,\\
&z_4(3) = 4z_1(2) - 6z_2(2) + 4z_3(2) - z_4(2) = 20C_1 - 34C_2 + 60C_3 - 90C_4,
\end{aligned}
\tag{6.66}
$$

where by applying the RTO Principle, we have assumed that $\bar{\sigma}_i(0) = 0, \bar{\sigma}_i(1) = 0,$ $\bar{\sigma}_i(2) = 0$ (until the next decision-time) which is equivalently to the constants C_i of the exogenous-input $z_i(0)$ remaining unchanged from the current time until the end of the process. Thus, at time k=0, the decision D_0 in (6.65) becomes

$$
D_0^* = -\frac{x(0) + z_1(0) + z_1(1) + z_1(2) + z_1(3)}{5} = -\frac{C_1 + C_2 + 2C_3 + 6C_4 + x(0)}{5}. \tag{6.67}
$$

At time k=1 and according to (6.63), the value of the exogenous-input $z_i(1)$ is determined to be

$$
\begin{aligned}
&z_1(1) = w_1 = C_1 + C_2 k + C_3 k^2 + C_4 k^3 = C_1 + C_2 + C_3 + C_4, \\
&z_2(1) = C_2 + 2C_3 k + 3C_4 k^2 = C_2 + 2C_3 + 3C_4, \\
&z_3(1) = 2C_3 + 6C_4 k = 2C_3 + 6C_4, \\
&z_4(1) = 6C_4, \\
&z_1(2) = z_2(1) = C_2 + 2C_3 + 3C_4, \\
&z_2(2) = z_3(1) = 2C_3 + 6C_4, \\
&z_3(2) = z_4(1) = 6C_4, \\
&z_4(2) = 4z_1(1) - 6z_2(1) + 4z_3(1) - z_4(1) = 4C_1 - 2C_2 + 4C_4, \\
&z_1(3) = z_2(2) = 2C_3 + 6C_4, \\
&z_2(3) = z_3(2) = 6C_4, \\
&z_3(3) = z_4(2) = 4C_1 - 2C_2 + 4C_4, \\
&z_4(3) = 4z_1(2) - 6z_2(2) + 4z_3(2) - z_4(2) = -4C_1 + 6C_2 - 4C_3 - 4C_4,
\end{aligned}
\tag{6.68}
$$

where the value of the exogenous-input $z_i(1)$ was determined by the implied composite-state observer and we have assumed $\bar{\sigma}_i(1) = 0, \bar{\sigma}_i(2) = 0$ (until the next decision-time) by applying the RTO principle, that is, we maintain the assumption that the constants C_i of the exogenous-input $z_i(1)$ will remain unchanged until the end of the process. Thus, at time k=1, the RTO decision D_1 in (6.65) becomes

$$
D_1^* = -\frac{x_1 + z_1(1) + z_1(2) + z_1(3)}{4} = -\frac{C_1 + 2C_2 + 5C_3 + 10C_4 + x(1)}{4}. \tag{6.69}
$$

At time k=2 and according to (6.63), the exogenous-input values are determined to be

$$
\begin{aligned}
&z_1(2) = w_2 = C_1 + C_2k + C_3k^2 + C_4k^3 = C_1 + 2C_2 + 4C_3 + 8C_4, \\
&z_2(2) = C_2 + 2C_3k + 3C_4k^2 = C_2 + 4C_3 + 12C_4, \\
&z_3(2) = 2C_3 + 6C_4k = 2C_3 + 12C_4, \\
&z_4(2) = 6C_4, \\
&z_1(3) = z_2(2) = C_2 + 4C_3 + 12C_4, \\
&z_2(3) = z_3(2) = 2C_3 + 12C_4, \\
&z_3(3) = z_4(2) = 6C_4, \\
&z_4(3) = 4z_1(2) - 6z_2(2) + 4z_3(2) - z_4(2) = 4C_1 + 2C_2 + 2C_4.
\end{aligned} \tag{6.70}
$$

Once again, we assumed that the values of the exogenous-input's constants C_i will remain at these new values until the end of the process and thus, $\bar{\sigma}_i(2) = 0$ and the RTO decision D_2 in (6.65) becomes

$$
D_2^* = -\frac{x(2) + z_1(2) + z_1(3)}{3} = -\frac{x(2) + C_1 + 3C_2 + 8C_3 + 20C_4}{3}. \tag{6.71}
$$

At k=3 and according to (6.63), the values of the exogenous-inputs are determined to be

$$
\begin{aligned}
&z_1(3) = w_3 = C_1 + C_2k + C_3k^2 + C_4k^3 = C_1 + 3C_2 + 9C_3 + 27C_4, \\
&z_2(3) = C_2 + 2C_3k + 3C_4k^2 = C_2 + 6C_3 + 27C_4, \\
&z_3(3) = 2C_3 + 6C_4k = 2C_3 + 18C_4, \\
&z_4(3) = 6C_4,
\end{aligned} \tag{6.72}
$$

and the RTO decision D_3 in (6.65) becomes

$$
D_3^* = -\frac{x(3) + z_1(3)}{2} = -\frac{x(3) + C_1 + 3C_2 + 9C_3 + 27C_4}{2}. \tag{6.73}
$$

Thus, at the end of the process at k=T=4, the RTO "optimal value" of the optimization criterion J becomes

$$J=(x_T)^2+\left[-\frac{x(0)+C_1+C_2+2C_3+6C_4}{5}\right]^2+\left[-\frac{x(1)+C_1+2C_2+5C_3+10C_4}{4}\right]^2$$
$$+\left[-\frac{x(2)+C_1+3C_2+8C_3+20C_4}{3}\right]^2+\left[-\frac{x(3)+C_1+3C_2+9C_3+27C_4}{2}\right]^2, \tag{6.74}$$

where x_T is the terminal state obtained at the end of the process.

Figure 6.15 shows the value of the optimization criterion for the cubic polynomial-spline behavior of w_k in (6.62) with three different initial state conditions x_0. The lines in Figure 6.15 were obtained by varying C_1 with $-20 \le C_1 \le 20$ and letting C_2=-5, C_3=-1.3 and C_4=1. The plus (+) line represents the value of J for the case in which x_0=0, the circle (o) line is for the case x_0=+20, and the dotted line is the case x_0=-20. Figure 6.15 also shows the optimization criterion value of Example #1. Thus, the gray dashed line represents the value of J for the baseline case with a zero exogenous-input as described in (6.10) for the case of x_0=±20.

Figure 6.15 illustrates the "Positive, Negative, and Zero Utility" regions of the cubic polynomial-spline type exogenous input when compared to the baseline case (no exogenous-input; zero value). By comparing the value of the optimization criterion J when there is no exogenous-input present (i.e., $w \equiv 0$ and $J=80$) to the value of J when the exogenous-input is a cubic polynomial-spline and with x_0=±20, one can observe that the "optimal" value of J is further minimized when the cubic polynomial-spline type exogenous-input takes values in the interval (-4, 3) for x_0=+20. For the case when x_0=-20, J is minimized further when the polynomial exogenous-input takes values in the interval (3, 11). This area of positive utility is depicted by the shaded region in Figure 6.15.

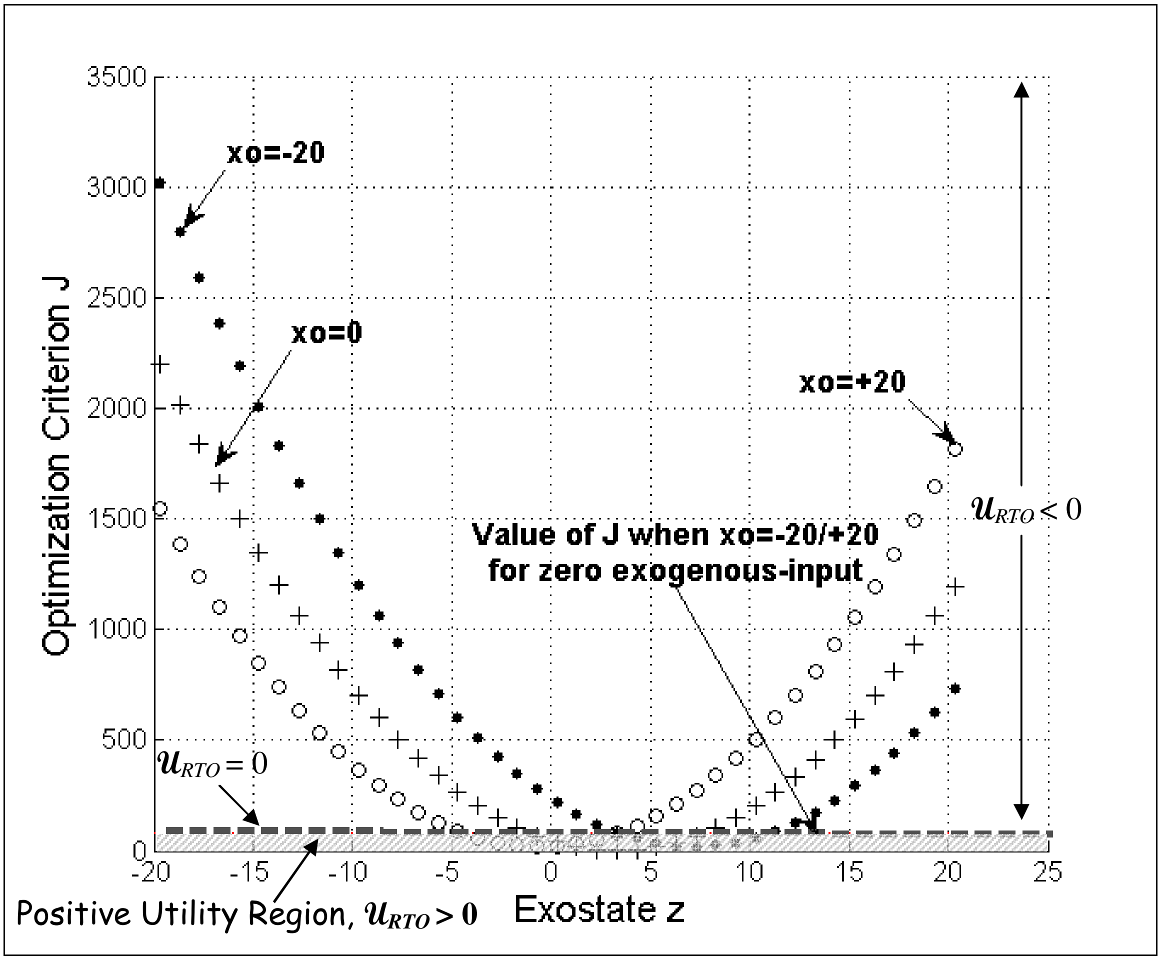

Figure 6.15. Criteria of Optimality and Utility Regions for Cubic Polynomial-Spline Type and Zero-Valued Exogenous-Inputs; $-20 \leq C_1 \leq 20$, $C_2=-5$, $C_3=-1.3$ and $C_4=1$ and Selected Fixed-Values of x_0

According to (6.51) and using (6.66), (6.68), (6.70) and (6.72), the value of the instantaneous utility $\mathcal{U}_k$ at each discrete-time has the same form (6.51) as in the previous Example (due to the nature of the matrix $H = [1 \quad 0 \quad 0 \quad 0]$) and is as follows

$$\begin{aligned}\mathcal{U}_0 &= -\frac{\left[z_1(0)+z_1(1)+z_1(2)+z_1(3)\right]\left[z_1(0)+z_1(1)+z_1(2)+z_1(3)+2x(0)\right]}{5} \\ &= -\frac{(2x(0)+C_1+C_2+2C_3+6C_4)(C_1+C_2+2C_3+6C_4)}{5}\end{aligned} \tag{6.75}$$

$$\begin{aligned}\mathcal{U}_1 &= -\frac{\left[z_1(1)+z_1(2)+z_1(3)\right]\left[2x(1)+z_1(1)+z_1(2)+z_1(3)\right]}{4} \\ &= -\frac{\left(C_1+2C_2+5C_3+10C_4\right)\left(2x(1)+C_1+2C_2+5C_3+10C_4\right)}{4}\end{aligned} \tag{6.76}$$

$$\begin{aligned}\mathcal{U}_2 &= -\frac{\left[z_1(2)+z_1(3)\right]\left[2x(2)+z_1(2)+z_1(3)\right]}{3} \\ &= -\frac{\left(C_1+3C_2+8C_3+20C_4\right)\left(2x(2)+C_1+3C_2+8C_3+20C_4\right)}{3}\end{aligned} \tag{6.77}$$

$$\begin{aligned}\mathcal{U}_3 &= -\frac{z_1(3)\left(2x(3)+z_1(3)\right)}{2} \\ &= -\frac{\left(C_1+3C_2+9C_3+27C_4\right)\left(2x(3)+C_1+3C_2+9C_3+27C_4\right)}{2}.\end{aligned} \tag{6.78}$$

Figure 6.16 shows the values of the utility in the ($\mathcal{U}$-x)-space for the specific Example #4 in which the exogenous-input z is a cubic polynomial-spline type that takes the values described by (6.62) and the state x only takes the values given by (6.1). The lines were obtained by determining the utility for the entire process k =0, 1, 2, 3 with the polynomial exogenous-input as described by (6.62) for the specific value of C_1=-20 and C_2=-5, C_3=-1.3 and C_4=1. Then, a new value was selected for C_1 (=-19) while C_2, C_3, and C_4 were held at the same specified values and the utility at each decision time computed again. This process was repeated for values of the polynomial exogenous input in which $-20 \le C_1 \le 20$. The plus (+) lines represent the value of the instantaneous

utility $\mathcal{U}$ for an initial state x_0=0, while the circle (o) lines correspond to x_0=+20 and the dotted lines correspond to x_0=-20.

From Figure 6.16 one can observe that the optimal state-value x_k progressively moves through regions of positive utility $\mathcal{U}_{RTO}$>0 and negative utility $\mathcal{U}_{RTO}$<0.

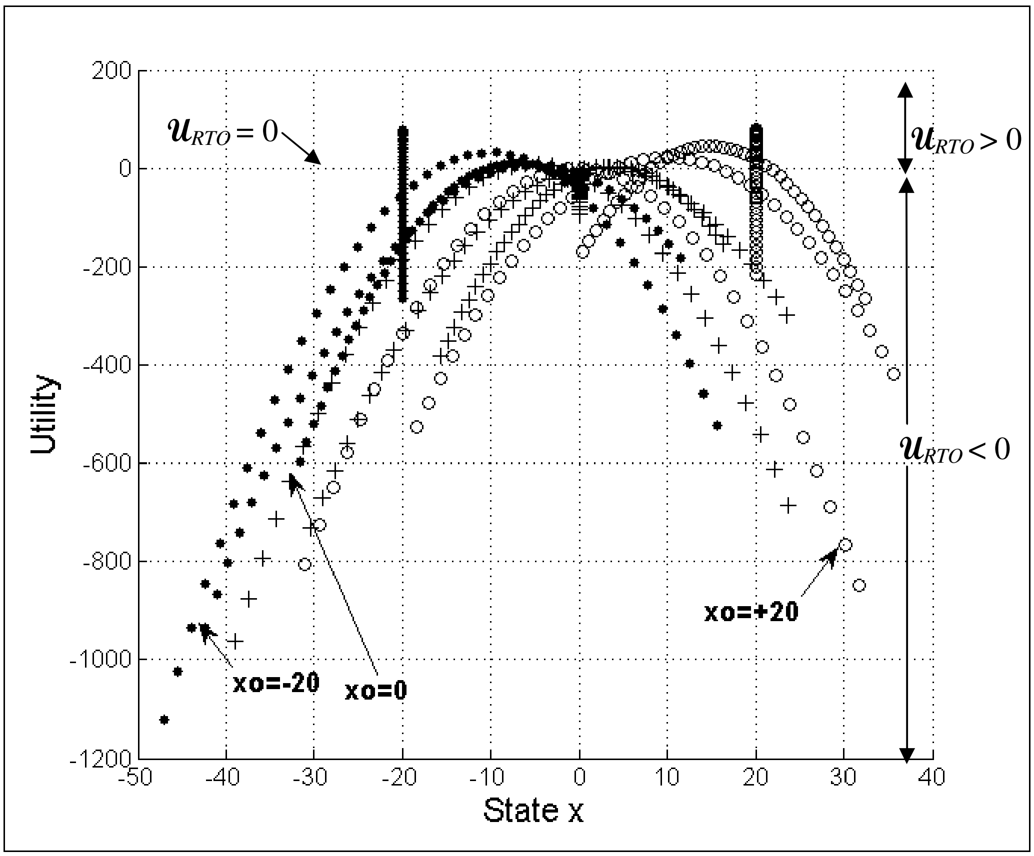

Figure 6.16. *x*-Axis Projection (View) of the Utility-Value $\mathcal{U}$ for the "Cubic Polynomial-Spline" Type Exogenous-Input in Example #4; 20≤C_1≤20, C_2=-5, C_3=-1.3 and C_4=1 and Selected Fixed-Values of x_0

Figure 6.17 illustrates the utility $\mathcal{U}$ for this Example in the ($\mathcal{U}$-z)-space indicated by the shaded regions. From Figure 6.17, it can be seen that the values of the cubic polynomial-spline type exogenous-input that give a positive utility $\mathcal{U}>0$ correspond to the same values ((-4, 3) for x_0=+20 and (3, 11) for x_0= -20) obtained from the Optimization Criterion figure (Figure 6.15) plus a larger region. This is due to the fact that the optimization criterion J is calculated at the end of the sequential decision process while the instantaneous utility $\mathcal{U}$ (as the name implies) is calculated at each discrete-time k. The cubic polynomial-spline type exogenous-input of this example contributes to *further minimizing* the "optimal"-value of the optimization criterion J in certain regions of the ($\mathcal{U}$-x-z)-space. This is a clear indication that certain values and time-behaviors of the exogenous-input of (6.62) acting on the state described by (6.1) and with and optimization criterion as described by (6.2) are in fact "useful" in further decreasing the value of the optimization criterion J for this example.

The domains of utility in the (x, z)-plane for this example at each discrete-time k depend on the value of the exogenous input state-variable $z_1(k)$ and the process state $x(k)$ and can be derived by setting the utility expression (6.51) equal to zero and solving for $z_1(k)$. The structure of the utility domains is the same as the utility domains (6.60) in the previous examples. Appendix E shows the derivation of such utility domains to obtain the following general form:

$$z_1(k)=0 \;\text{ and }\; z_1(k)=2Kx(k);\; K=-\frac{a^{(N-k)}}{\sum_{j=k}^{N-1}(a^j f)}, \tag{6.79}$$

where K is a scalar quantity which varies with each discrete time k, a is the sequential process scalar coefficient in (6.1), f is the exogenous-input coefficient in (6.1), and N is the number of stages for the entire sequential-decision process ($N = T = 4$). The utility domains look the same as in the previous example and are already depicted in Figures (6.12) and (6.6). Expression (6.79) is a general utility expression that can be applied to an n^{th}-order exostate of a structured-variation type exogenous-input $w(k)$ when the sequential-decision process is of the form (6.1) (x is a scalar), and the criterion of optimality has the form in (6.2).

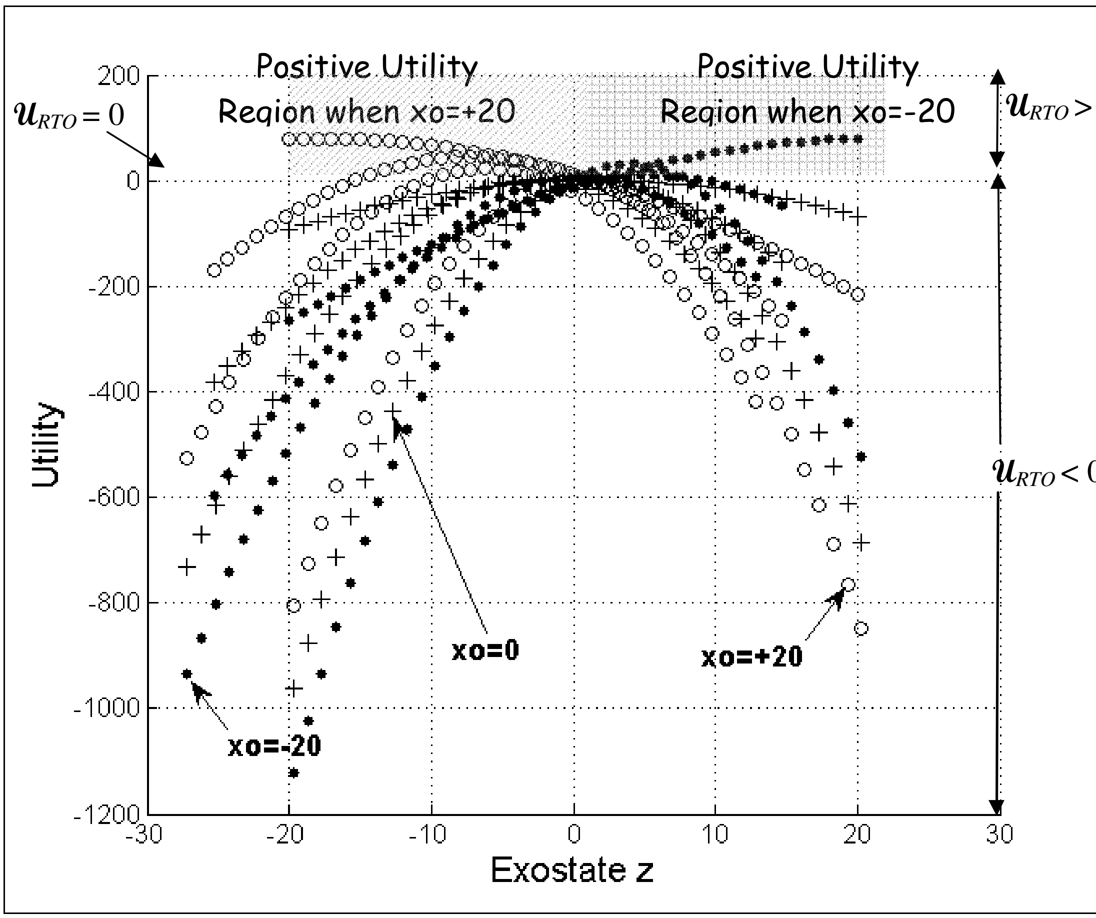

Figure 6.17. z-Axis Projection (View) of Utility-Value $\mathcal{U}$ for the "Cubic Polynomial-Spline" Type Exogenous-Input of Example #4; $20 \leq C_1 \leq 20$, C_2=-5, C_3=-1.3 and C_4=1 and Selected Fixed-Values of x_0

The effectiveness of the cubic-polynomial-spline type "exogenous-input-utilizing decision" is based on expression (5.24) and is shown in Figure 6.18. Values for the effectiveness $\mathcal{E}$ were computed for the current example in which the exogenous-input is a cubic polynomial-spline type and plotted against the values of z. Just as in the previous examples, the values of z which give a positive effectiveness $\mathcal{E} > 0$ correspond to the same values of z that further minimized the optimization criterion J in Figure 6.15. The highest values of the effectiveness for this example are 77.3% that occurs when x_0=+20 and z = -0.7 and 75.3% that occurs when x_0=-20 and z=7.3. The effectiveness $\mathcal{E}$ of the "exogenous-input utilizing decision" (EIU-"smart decision") is surprisingly high for this example which is not apparent from inspection of the criterion of optimality plot in Figure 6.15.

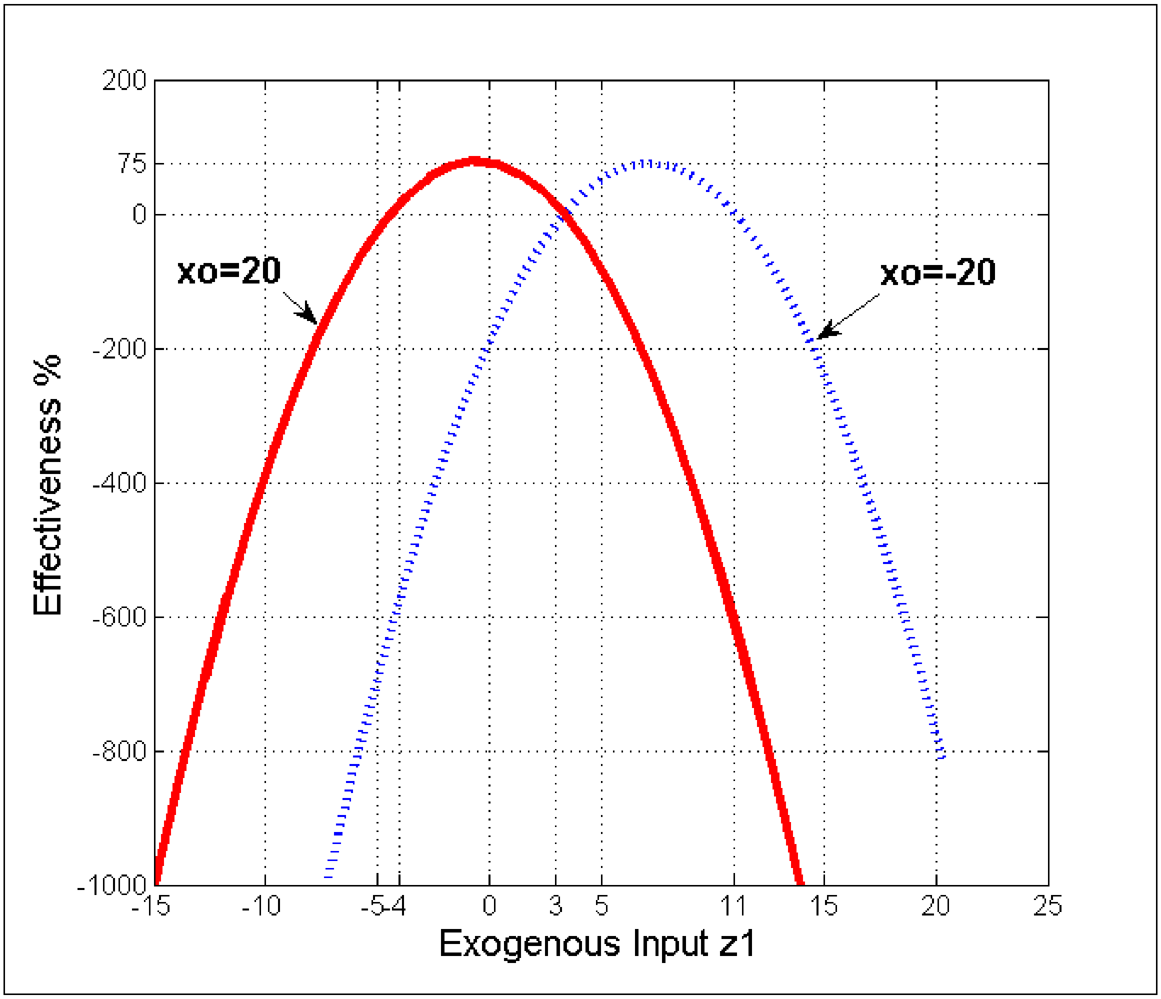

Figure 6.18. Values of Effectiveness $\mathcal{E}$ for Example #4 for x_0=±20

Chapter 7

SUMMARY, CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

This dissertation has addressed the class of Sequential-Decision Processes with uncertain exogenous-inputs in which the objective is to make an ordered sequence of "decisions" to optimize a certain mathematical expression, representing a performance metric for the outcome of the decisions, where the underlying process is subject-to a "time-sparse Kronecker delta function" driven exogenous-input dynamics. These types of uncertain exogenous-inputs, known as exogenous-inputs with *structured-variations* do not satisfy the Ergodic hypothesis, cannot be characterized in terms of long-term statistical properties, and have distinct patterns of qualitative time-behavior over short periods of time.

Using Johnson's state-variable type modeling procedure [30]-[33] to characterize the uncertain time-behavior of structured-variation type exogenous-inputs, we showed how such exogenous-inputs can be incorporated-into the Dynamic Programming Method of Bellman where the 'state" (exostate) of the structured-variation type exogenous-input can be estimated (by means of a real-time Kalman filter or state observer [42]) along with the underlying SDP process state. In this way, we confirmed Johnson's assertion [38]

that when exogenous-inputs with structured-variations are incorporated into optimal sequential-decision processes using the Dynamic Programming methodology, the "state" Bellman refers-to in his Principle of Optimality cannot be "known" a-priori, and thus, is not a valid "state" for the Dynamic Programming backward-stepping solution process. This fact makes it impossible to employ Bellman's solution procedure associated with the Dynamic Programming methodology. However, when we invoked the more general Principle of "Real-Time Optimality" (RTO), introduced in [38], and set up a "new" composite state consisting of the process state and the exogenous-input exostate ($x|z$), we were able to apply Bellman's backward-solution procedure to the Dynamic Programming methodology to obtain a Real-Time Optimal (RTO) decision which is the best possible, rational decision that can be made in the presence of uncertain, uncontrollable, unmeasurable exogenous-inputs that have structured-variations with time-sparse jumps. The RTO decisions thus obtained are not optimal in the absolute sense but can only be improved by fortuitous "gambling" about the "strength" and arrival times of sparse-in-time, isolated events that may or may not occur in the future and are thus non-Ergodic events which are unknown to the decision-maker. The gambling about such time-sparse events in SDP-type problems can involve treacherous situations.

In this dissertation, we have applied the Structured-Variation methodology and the RTO principle to some specific, analytic, first-order multistage sequential-decision processes with structured-variation-type exogenous-inputs and a specific class of performance-criteria. In particular, the general form for the RTO-optimal decision as a function of the "real-time" process state, and the exogenous-input state, at each decision-time t_k has been developed, using a variation of the traditional Dynamic Programming

Method. During this process, we were able to show, by analytic means, that in certain cases, the exogenous-inputs can have a positive effect or "positive utility" by reducing the value of the optimization criteria even further than if no exogenous-inputs were present in the sequential-decision process. The calculation of a general expression for the utility of a first-order *N*-step sequential-decision process was illustrated. The unorthodox idea of the "positive utility" of an exogenous-input introduced in [35] has been conclusively demonstrated in the specific examples which were presented in this dissertation, along with their specific domains of utility in the corresponding (x-z)-space.

7.1 Contributions of this Dissertation

In summary, the contributions of this dissertation are

1. Extension of the uncertain exogenous-input modeling procedure, as traditionally used in sequential-decision processes to represent uncertain inputs, to include a class of uncertain "structured-variation" type exogenous-inputs that are <u>not</u> Markovian processes, are non-Ergodic, and cannot be effectively characterized by long-term statistical averages, in general.

2. Integration of "structured-variation" type uncertain exogenous-inputs, and associated state-models into the traditional Dynamic Programming methodology for solving a general class of optimal sequential-decision-problems involving a class of structured-variation type exogenous-inputs.

3. Incorporation of the Principle of Real-Time Optimality (RTO) into the modified Dynamic Programming solution method in #2 for obtaining RTO-optimal decisions in sequential-decision problems.

4. Development and analytical solution of representative numerical-examples that demonstrate, with explicit numerical proofs, the fact (apparently never before recognized or studied in the SDP literature) that the systematic, new "sequential decision-algorithms" proposed here can produce "extraordinarily-optimal" decisions in the sense that the decisions can be designed to achieve maximum "harvesting" of the useful-effects of uncertain exogenous-inputs, i.e., make maximal-use of uncertain exogenous-inputs to (i) further optimize the process performance beyond the optimal performance obtainable when there are no exogenous-inputs, or (ii) minimize the inevitable loss of process-performance due to the presence of exogenous-inputs that, by their nature, can only reduce the process-performance (i.e., have only negative "utility"). Thus, the new SDP decision-algorithm proposed here can be characterized as producing "win-win" optimal-decisions (in the RTO sense) in those cases where uncertain, structured-variation type exogenous-inputs act-on the underlying process.

7.2 Recommendations for Future Work

Based on the research presented in this dissertation, several directions for further work, in the area of sequential-decision processes with exogenous-inputs, have emerged. One of those directions is the application of the methods used here to more complex-processes such as 2^{nd} or 3^{rd}-order, or processes that have different dynamics at different "stages" to further understand the benefits of the methods (and associated computational burdens) for more complex processes. Another obvious direction for further study is the selection of other types of performance criteria J (such as exponential cost functions [49], [29]) to maximize "utility" of exogenous-inputs as well as the investigation of more

complex forms of structured-variation-type exogenous-inputs and their influence on the domains of utility. Another interesting direction for further research is the incorporation of the new class of “sequential-decisions” involving non-constant interstage behaviors of the decisions, to improve the optimal performance of decisions in SDPs. That is, application of the structured-variation “exogenous-input utilizing” decisions introduced here for the new class of a “discrete-continuous” (D/C)-type sequential decision [36] in which each sequential decision is allowed to vary continuously across the sample interval *k*, in a “smart,” pre-determined manner to (possibly) achieve “more-optimal” SDP decisions, in general, and/or more thorough (optimal) utilization of uncertain exogenous-inputs in SDPs.

APPENDICES

APPENDIX A

DERIVATION OF THE GENERAL EXPRESSION FOR THE RTO-DECISION OF AN *N*-STAGE, FIRST-ORDER, SDP

In Chapter 4, we considered a process modeled by a first order, discrete-time state-evolution equation which describes how the "current" state of the underlying dynamic process (and the effect of the "current" exogenous-input) alter the sequential transition of the process state x from one stage (k) to the next stage ($k+1$). That process state-model is assumed to have the form of the general, linear first-order difference-equation

$$x_{k+1} = a_k x_k + b_k D_k + f_k w_k \,, \tag{A.1}$$

where the sequential "stages" (indexed by k) in the sequential decision problem extend over the finite time interval $t_0 \leq k \leq T,\ T{=}fixed < \infty$, x_k is the scalar state, D_k is the scalar decision determined at each decision-time k, w_k is an unknown, unmeasurable, and uncontrollable scalar exogenous-input, and the coefficients (a, b, and f) are also scalar and known, arbitrary real functions, which, in principle, can change with time in some

manner (assumed known a priori). To simplify the calculations in Chapter 4, we considered the *time-invariant* case (a, b, and f = known constants).

The uncertain time-behavior of the scalar exogenous-input w_k, was modeled in Chapter 4 by the structured-variation technique, leading to the following first-order, discrete-time, exostate-model

$$\begin{aligned} w_k &= hz_k; \quad (h = 1.0) \\ z_{k+1} &= gz_k + \overline{\sigma}_k \ , \end{aligned} \tag{A.2}$$

where z_k denotes the state of the exogenous-input w_k, the coefficients h and g denote arbitrary, real, known constants chosen by the decision maker and $\overline{\sigma}_k$ is a vector of *time-sparse* sequences of Kronecker delta functions with completely unknown arrival times and values [34] .

The scalar-valued criterion of optimality J to be *minimized* for the sequential decision process (A.1) was chosen to have the following specific form

$$J = x_T{}^2 + \sum_{k=t_o}^{N-1} (D_k)^2 \ , \tag{A.3}$$

where the first term on the right-side of the optimization expression (A.3) is placed there to penalize deviations of the terminal-state value x_T from the (presumed) desired value $x_T = 0$ while the second term on the right-side of (A.3) discourages excessive magnitude/intensity of the decisions D_k.

In Chapter 4, we also developed the discrete time state-evolution model for the composite state $\tilde{x} \underline{\underline{\Delta}}\, col.(x|z)$ to be

$$\tilde{x}_{k+1} = \begin{bmatrix} x_{k+1} \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} a & fh \\ 0 & g \end{bmatrix} \cdot \begin{bmatrix} x_k \\ z_k \end{bmatrix} + \begin{bmatrix} b \\ 0 \end{bmatrix} \cdot D_k + \begin{bmatrix} 0 \\ \sigma_k \end{bmatrix}; \; \tilde{x} \underline{\underline{\Delta}} \left(\frac{x}{z} \right), \tag{A.4}$$

where x is the scalar state process, $k = t_o, \ldots, (N\text{-}1)$, $N = T$ and z_k is the "current" value of the "exostate" z of the structured-variation type exogenous-input w_k, as modeled by (A.2).

By using the backward-stepping Dynamic Programming technique described in Chapters 3 and 4, and starting at the last decision state t_k=k=(N-1), we solved for the RTO decision that gives a minimum value of J for an *arbitrary* value of the process state $x_{N\text{-}1}$ and an *arbitrary* value of the "exostate" $z_{N\text{-}1}$. Thus, the RTO-optimal decision D^*_{N-1} at stage k=(N-1) is the scalar decision (see derivation of (4.8) in Section 4.1)

$$D^*_{N-1} = -\frac{b(ax_{N-1} + fhz_{N-1})}{1+b^2}. \tag{A.5}$$

At k= (N-2) the scalar RTO-optimal decision is

$$D^*_{N-2} = -\frac{ab\left[a^2 x_{N-2} + afhz_{N-2} + fhz_{N-1}\right]}{1+b^2+(ab)^2} = -\frac{ab\left[a^2 x_{N-2} + afhz_{N-2} + fhgz_{N-2}\right]}{1+b^2+(ab)^2}, \tag{A.6}$$

where the equation on the right of the second equal sign is written in terms of only (x_{N-2}, z_{N-2}) by letting $z_{N-1} = gz_{N-2}$ as defined in (A.2) and (A.4).

At k= (N-3) the scalar RTO-optimal decision is

$$D_{N-3}^{*} = -\frac{a^2 b\left[a^3 x_{N-3} + a^2 fhz_{N-3} + afhz_{N-2} + fhz_{N-1}\right]}{1 + b^2 + (ab)^2 + (a^2 b)^2} = -\frac{a^2 b\left[a^3 x_{N-3} + a^2 fhz_{N-3} + afhgz_{N-3} + fhg^2 z_{N-3}\right]}{1 + b^2 + (ab)^2 + (a^2 b)^2}, \tag{A.7}$$

where once again the second equation is written in terms of only z_{N-3} by use of (A.2).

At k= (N-4) the scalar RTO-optimal decision is

$$D_{N-4}^{*} = -\frac{a^3 b\left[a^4 x_{N-4} + a^3 fhz_{N-4} + a^2 fhz_{N-3} + afhz_{N-2} + fhz_{N-1}\right]}{1 + b^2 + (ab)^2 + (a^2 b)^2 + (a^3 b)^2} = -\frac{a^3 b\left[a^4 x_{N-4} + a^3 fhz_{N-4} + a^2 fhgz_{N-4} + afhg^2 z_{N-4} + fhg^3 z_{N-4}\right]}{1 + b^2 + (ab)^2 + (a^2 b)^2 + (a^3 b)^2}. \tag{A.8}$$

A pattern for the decision at each decision stage k can be distinguished by inspecting the decision equations (A.5)-(A.8). From this inspection, we arrive at a general RTO optimal decision $D_k^* = D_k^*(x_k, z_k, k)$ for the N-stage first-order sequential-decision process of (A.1)-(A.3) which is the following scalar expression

$$D_k^{*} = -\left[\frac{a^{(N-1-k)} b}{1 + \sum_{j=k}^{N-1} a^{2(N-1-j)} b^2}\right] \times \left[a^{(N-k)} x_k + \sum_{j=k}^{N-1} fh\left(a^{(N-1-j)}\right) z_j\right]. \quad k = 0, 1, 2, \ldots, N-1. \tag{A.9}$$

APPENDIX B

DERIVATION OF THE GENERAL EXPRESSION FOR THE VALUE-FUNCTION $\mathcal{V}_{RTO}$ CORRESPONDING-TO THE RTO MINIMIZATION OF THE *N*-STAGE, FIRST-ORDER LINEAR SDP (EQUATION 5.15)

In Chapter 5, we considered an *N*-stage first-order linear SDP. That process state-model is assumed to have the form of the general, linear first-order difference-equation

$$x_{k+1} = a_k x_k + b_k D_k + f_k w_k \,, \tag{B.1}$$

where the sequential "stages" (indexed by k) in the sequential decision problem extend over the finite time interval $t_0 \leq k \leq T$, $T = fixed < \infty$, x_k is the scalar state, D_k is the scalar decision determined at each decision-time k, w_k is an unknown, unmeasurable, and uncontrollable scalar exogenous-input, and the coefficients (a, b, and f) are also scalar and known, arbitrary real functions, which, in principle, can change with time in some

manner (assumed known a priori). To simplify the calculations, we considered the *time-invariant* case (a, b, and f = known constants).

The uncertain time-behavior of the scalar exogenous-input w_k, was modeled in Chapter 4 by the structured-variation technique, leading to the following first-order, discrete-time, exostate-model

$$\begin{aligned} w_k &= hz_k; \quad (h = 1.0) \\ z_{k+1} &= gz_k + \overline{\sigma}_k \ , \end{aligned} \tag{B.2}$$

where the coefficients h and g denote arbitrary, real, known constants chosen by the decision maker.

The scalar "value-function" (payoff-function) $\mathcal{V}$ associated-with the first-order SDP of (B.1)-(B.2) is the "holy-grail" of the Hamilton-Jacobi approach [18] to optimization theory and is here defined as the minimum value of J obtained with the Dynamic Programming solution method when structured-variation type exogenous-inputs of the form (B.2) are present in a SDP defined by (B.1) and the following (corresponding) RTO-optimal decision is employed

$$D_k^* = -\left[\frac{a^{(N-1-k)}b}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)}b^2}\right] \times \left[a^{(N-k)}x_k + \sum_{j=k}^{N-1} fh\left(a^{(N-1-j)}\right)z_j\right]. \ \ k = 0,1,2,\ldots,N-1. \tag{B.3}$$

To derive the expression for $\mathcal{V}$ for the SDP described by (B.1) and (B.2) we consider the following general linear-quadratic optimization criterion using the RTO decision (B.3) at each decision stage k= (N-1), (N-2), …, 1, 0

$$J = \frac{1}{2} x_N^T S x_N + \frac{1}{2} \sum_{k=0}^{N-1} \left(x_k^T Q_k x_k + D_k^T R_k D_k \right). \tag{B.4}$$

In (B.4), S, Q, and R are positive definite, symmetric matrices chosen by the designer, N is the specified terminal time, and $(.)^{\mathrm{T}}$ indicates the transpose of (.). The presence of the "decision penalty term $D_k^T R_k D_k$" under the summation sign in J encourages the sequential-decisions to strive-to make maximum utilization of any "free" energy or process "enhancing effects," embodied-in the perturbing effects of the exogenous-input w_k on the process-behavior, to reduce or minimize the imposition of <u>excessively</u> severe (large) "decision-values" D_k. For this SDP described by (B.1)-(B.2), we will assume R=1, Q=0, and S=1. Thus, J has the following form

$$J = {x_N}^2 + \sum_{k=t_o}^{N-1} (D_k)^2 \quad . \tag{B.5}$$

Substituting the RTO optimal decision (B.3) into (B.5), we obtain the RTO Value-Function $\mathcal{V}_{RTO}$ for the RTO optimization of (B.1), (B.2), and (B.5) for the decision stages (N-1), (N -2), (N -3), and (N -4) as follows

$$\begin{aligned}
\mathcal{V}_{\mathrm{RTO(N-1)}} &= (a^2 x_{N-1}^2 + 2afhx_{N-1} z_{N-1} + f^2 h^2 z_{N-1}^2) \big/ (1+b^2) \\
\mathcal{V}_{\mathrm{RTO(N-2)}} &= (a^4 x_{N-2}^2 + 2a^2 fh(a+g) x_{N-2} z_{N-2} + f^2 h^2 (a+g)^2 z_{N-2}^2) \big/ (1+b^2+a^2b^2) \\
\mathcal{V}_{\mathrm{RTO(N-3)}} &= \frac{(a^6 x_{N-3}^2 + 2a^3 fh(a^2+ag+g^2) x_{N-3} z_{N-3} + f^2 h^2 (a^2+ag+g^2)^2 z_{N-3}^2)}{(1+b^2+a^2b^2+a^4b^2)} \\
\mathcal{V}_{\mathrm{RTO(N-4)}} &= \frac{(a^8 x_{N-4}^2 + 2a^4 fh(a^3+a^2g+ag^2+g^3) x_{N-4} z_{N-4} + f^2 h^2 (a^3+a^2g+ag^2+g^3)^2 z_{N-4}^2)}{(1+b^2+a^2b^2+a^4b^2+a^6b^2)},
\end{aligned} \tag{B.6}$$

where a, b and f are the scalar, known, arbitrary real functions as defined in (B.1) and h and g are the arbitrary, real, known, constant coefficient defined in (B.2) as part of the exogenous-input exostate model.

Inspection of the interior terms of (B.6) (inside parentheses), shows an emerging pattern. Thus, for decision stage (N-1), the interior terms in parentheses are

$$(a+g)^0 = 1 . \tag{B.7}$$

For decision stage (N-2), the interior terms in parentheses are

$$(a+g) . \tag{B.8}$$

For decision stage (N-3), the interior terms in parentheses are

$$(a^2 + ag + g^2) . \tag{B.9}$$

For decision stage (N-4), the interior terms in parentheses are

$$(a^3 + a^2 g + ag^2 + g^3) . \tag{B.10}$$

The form of the terms in (B.7)-(B.10) resemble the Binomial theorem in which the expansion of $(a+g)^n$ for a few values of n (a positive integer) has the following form

$$\begin{aligned}
&(a+g)^0 = 1 \\
&(a+g)^1 = a+g \\
&(a+g)^2 = a^2 + 2ag + g^2 \\
&(a+g)^3 = a^3 + 3a^2 g + 3ag^2 + g^3 \\
&(a+g)^4 = a^4 + 4a^3 g + 6a^2 g^2 + 4ag^3 + g^4 .
\end{aligned} \tag{B.11}$$

The only difference between the terms in (B.7)-(B.10) and the expansion terms of the Binomial theorem in (B.11) is that the interior coefficients of $a^{n-m} g^m$ are all equal to one

where n and m are positive integers. The Binomial coefficient of the interior term $a^{n-m}g^{m}$ is given by [45]

$$ {}_{n}C_{m} = \frac{n!}{m!(n-m)!} = \frac{n(n-1)(n-2)\ldots(n-m+1)}{m!} . \tag{B.12} $$

Thus, we can define a new term $\overline{(a+g)}^{(N-k)}$ in which all the coefficients of the interior terms $a^{n-m}g^{m}$ are equal to one as

$$ \overline{(a+g)}^{n} \triangleq a^{n} + \frac{n}{n}a^{n-1}g + \ldots + \frac{{}_{n}C_{m}}{{}_{n}C_{m}}a^{n-m}g^{m} + \ldots + \frac{n}{n}ag^{n-1} + g^{n} , \tag{B.13} $$

where the Binomial coefficient ${}_{n}C_{m}$ of $a^{n-m}g^{m}$ is given by (B.12).

Thus, the results in (B.6) can be generalized to the following scalar expression

$$ \mathcal{V}_{RTO}(k) = \left(\frac{a^{2(N-k)}}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)}b^{2}} \right) x_{k}^{2} + 2\left(\frac{a^{(N-k)} fh\overline{(a+g)}^{(N-k)}}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)}b^{2}} \right) x_{k} z_{k} + \left(\frac{(fh)^{2}\overline{(a+g)}^{2(N-k)}}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)}b^{2}} \right) z_{k}^{2} \tag{B.14} $$

where the term $\overline{(a+g)}^{(N-k)}$ is defined as in (B.13).

APPENDIX C

MATLAB CODE

DRIVER.m

```
% Driver for plant x(n+1)=x(n)+f(n)*D(n)+w(n) and Optimization
criterion
% J =[x(T)]^2 + sum(D(n)^2)

% w is a vector with disturbance values
% f is a vector representing a known function that scales the Decision
% and the same size as w
% xo is a scalar initial condition

close all
clear all
type = 'polynomial';% 'constant' disturbance; 'ramp' disturbance;
'polynomial'
f = [1,1,1,1];
counter = 1;

for xo=-20:20:20

    switch lower(type)
        case 'constant'
            % Constant Slope Disturbance
            % Note: uses the same Supreme OC in backward time but with
a
            % constant slope disturbace
            % Define disturbance
            %         %*** CONSTANT STEP ***
            h=1;
            g=1;
            %*** CONSTANT STEP ***
            C2 = 0;
            i=1;
            %C1 = 0;
            for C1=-20:20
                % constant-step
```

```
                ws = [C1 C1 C2 C2];

                % RTO in Forward Time
                % Note: for utility studies, change wr = wc
                wr = ws;
                [Jrto,xr,zr,Jr,D,util] =
RTOModelForward(wr,h,'constant',f,xo,C2,g);
                dist(i,:) = zr;
                utility(i,:) = util;
                n=length(util);
                if(i==1)
                    h1=figure(1);hold on;
                    h2=figure(2);hold on;
                end
                if(xo==-20)

figure(h1);plot3(xr(1:n),zr(1:n),util,'.k','MarkerSize',14);

figure(h2);plot3(xo,zr(1,1),Jr,'.k','MarkerSize',14);
                elseif(xo==20)
                    figure(h1);plot3(xr(1:n),zr(1:n),util,'ok');
                    figure(h2);plot3(xo,zr(1,1),Jr,'ok');
                elseif(xo==0)
                    %                 ind=find(util>0); % find positive
utility and plot in red
                    %
figure(h1);plot3(xr(1:n),zr(1:n),util,'+r','MarkerSize',8);
                    %
plot3(xr(ind),zr(ind),util(ind),'+r','MarkerSize',8);

figure(h1);plot3(xr(1:n),zr(1:n),util,'+k','MarkerSize',8);

figure(h2);plot3(xo,zr(1,1),Jr,'+k','MarkerSize',8);
                end
                % Save Joptimal for computation of Effectiveness
                constJ(i,counter) = Jr;
                dist(i,counter)=zr(1,n+1);
                i= i+1;
            end
            counter = counter+1;
            figure(h1)
            xlabel('State x','FontSize',16);
            ylabel('Exostate z','FontSize',16);
            zlabel('Utility','FontSize',16)
            grid
            hold on;
            figure(h2)
            xlabel('State x','FontSize',16);
            ylabel('Exostate z','FontSize',16);
            zlabel('Optimization Criterion J','FontSize',16);
            grid
            hold on;
        case 'ramp'
            % Constant Slope Disturbance
            % Note: uses the same Supreme OC in backward time but with
a
            % constant disturbace
```

```
            % Define disturbance
            %*** RAMP ***
            C2 = 2;
            i=1;
            for C1=-20:20
                %             % Define Disturbance
                %             %*** RAMP ***
                H = [1 0];
                G = [0 1;-1 2];

                %*** RAMP ***
                % ramp
                ws = [C1 C1+C2 C1-2*C2 C1-3*C2];
                wr = ws;
                Cons = [C1 C2];
                [Jrto,xr,zr,Jr,D,util] =
RTOModelForward(wr,H,'ramp',f,xo,Cons,G);
                utility(i,:) = util;
                %Plot Utility
                n=length(util);
                if(i==1)
                    h1=figure(1);hold on;
                    h2=figure(2);hold on;
                end
                if(xo==-20)
%
figure(h1);plot3(xr(1:n),ws,util,'.k','MarkerSize',14);
%
figure(h2);plot3(xo,ws(4),Jr,'.k','MarkerSize',14);

figure(h1);plot3(xr(1:n),zr(1:n),util,'.k','MarkerSize',14);

figure(h2);plot3(xo,zr(1,n+1),Jr,'.k','MarkerSize',14);
                elseif(xo==20)
%                     figure(h1);plot3(xr(1:n),ws,util,'ok');
%                     figure(h2);plot3(xo,ws(4),Jr,'ok');
                    figure(h1);plot3(xr(1:n),zr(1:n),util,'ok');
                    figure(h2);plot3(xo,zr(1,n+1),Jr,'ok');
                elseif(xo==0)
%
figure(h1);plot3(xr(1:n),ws,util,'+k','MarkerSize',8);
%
figure(h2);plot3(xo,ws(4),Jr,'+k','MarkerSize',8);

figure(h1);plot3(xr(1:n),zr(1:n),util,'+k','MarkerSize',8);

figure(h2);plot3(xo,zr(1,n+1),Jr,'+k','MarkerSize',8);
                end
                %Plot J for J(xo=20,-20) in optimization plot
                 figure(h2);plot3(20,C1,(20^2)/5,'k','MarkerSize',14);
                % Save Joptimal for computation of Effectiveness
                constJ(i,counter) = Jr;
                dist(i,counter)=zr(1,n+1);
                i= i+1;
            end
            counter = counter+1;
            figure(h1)
```

```
            xlabel('State x','FontSize',16);
            ylabel('Exostate z','FontSize',16);
            zlabel('Utility','FontSize',16)
            grid
            hold on;
            figure(h2)
            xlabel('State x','FontSize',16);
            ylabel('Exostate z','FontSize',16);
            zlabel('Optimization Criterion J','FontSize',16);
            grid
            hold on;

        case 'polynomial'
            %*** POLYNOMIAL ***
            i=1;
            for C=-20:20
                % Define Disturbance
                %*** POLYNOMIAL ***
                H = [1 0 0 0];
                G = [0 1 0 0
                    0 0 1 0
                    0 0 0 1
                    4 -6 4 -1];
                t=0:3; % discrete time
                C1=C;%3;
                C2=-5;
                C3=-1.3;
                C4 = 1;
                w=C1 + C2*(t) + C3*(1*t).^2 + C4*(t).^3;
                Cons = [C1 C2 C3 C4];
                [Jrto,xr,zr,Jr,D,util] =
RTOModelForward(w,H,'polynomial',f,xo,Cons,G);
                utility(i,:) = util;
                %Plot Utility
                n=length(util);
                if(i==1)
                    h1=figure(1);hold on;
                    h2=figure(2);hold on;
                end
                if(xo==-20)

figure(h1);plot3(xr(1:n),zr(1,1:n),util,'.k','MarkerSize',14);
                    %Plot Jr with last value of z (at k=4), not the
                    %predicted z at k=5

figure(h2);plot3(xo,zr(1,n),Jr,'.k','MarkerSize',14);
                elseif(xo==20)
                    figure(h1);plot3(xr(1:n),zr(1,1:n),util,'ok');
                    figure(h2);plot3(xo,zr(1,n),Jr,'ok');
                elseif(xo==0)

figure(h1);plot3(xr(1:n),zr(1,1:n),util,'+k','MarkerSize',8);

figure(h2);plot3(xo,zr(1,n),Jr,'+k','MarkerSize',8);
                end
                % Save Joptimal for computation of Effectiveness
                constJ(i,counter) = Jr;
```

```
                dist(i,counter)=zr(1,n);
                i= i+1;
            end
            counter = counter+1;
            figure(h1)
            xlabel('State x','FontSize',16);
            ylabel('Exostate z','FontSize',16);
            zlabel('Utility','FontSize',16)
            grid
            hold on;
            figure(h2)
            xlabel('State x','FontSize',16);
            ylabel('Exostate z','FontSize',16);
            zlabel('Optimization Criterion J','FontSize',16);
            grid
            hold on;
        otherwise
            disp('Unknown method.')
    end

end
%Plot J for J(xo=20,-20) in optimization plot
figure(h2);plot3(20,-20:20,(20^2)/5,'--r','MarkerSize',14);
% Effectivenes
h3=figure(3);hold on;
i=1;
for x0=-20:20:20
    Jnd=((x0).^2)/5;
    E(:,i)=100*(Jnd-constJ(:,i))./Jnd;
    i=i+1;
end
% figure(h3);plot(-20:20,E)
figure(h3);plot(dist,E)
xlabel('Exogenous Input, z')
ylabel('Effectiveness')
createEffectivenessfigure(dist, E)
```

RTOMODELFORWARD.m

```
function [Jrto,x,z,Jo,D,util2] = RTOModelForward(w,H,type,f,xo,B2,G)
% Real Time Optimal Controller In Forward Time
% Equation 19 of my paper titled "Utility of Exogenous-Inputs in
Dynamic
% Programming"

% Define/Initialize variables
term1den = 0;
term2 = 0;
x(1) = xo;
sumD(1)=0;
a=1;
b=1;
assist = 0;
burden = 0;
Rden = 0;
A2 = 0;
Bur = 0;
R=0;

% Number of samples
N = length(w);

for k=1:N %Matlab can't use an index of zero so I'm starting at 1

    % In Reality, we don't know what the disturbance will be in the
future
    % so fix the disturbance and use the Decision D(k) obtained with
the
    % constant slope case (assuming disturbance remains at a constant
slope for all
    % time). When the disturbance changes, then we
    % will adjust our decision at that time.

    % k has to be zero for equation 19 to work accurately. Use a dummy
    % variable kz = k-1
    kz = k-1;

    % Define Exogenous-Input state-space model
    if strcmp(type,'ramp')
        z(:,k) = [w(k); B2(1)+(kz+1)*B2(2)];
        if (k>=3)
            z(:,k) = [w(k); B2(1)-(kz+1)*B2(2)];
        end
        z(:,k+1) = G*z(:,k);
    elseif strcmp(type,'constant')
        % Define Exogenous-Input state-space model
        z(:,k) = w(k);
        z(:,k+1) = G*z(k);
    elseif strcmp(type,'polynomial')
        % B2 = [C1 C2 C3 C4]
        C2= B2(2);
```

```
        C3= B2(3);
        C4= B2(4);
        z(:,k) = [w(k); C2+2*C3*kz+3*C4*kz^2; 2*C3+6*C4*kz; 6*C4];
        disturbance = H*z(:,k);
        z(:,k+1) = G*z(:,k);
    end

   % Determine Decision
    term1num = (a^(N-1-kz))*b;
    for j=kz:N-1
        if (j>kz+1) % define future z
            if strcmp(type,'ramp')
                z(:,j+1) = G*z(:,j);
            elseif strcmp(type,'constant')
                z(j+1) = z(j);
            elseif strcmp(type,'polynomial')
                z(:,j+1) = G*z(:,j);
            end
        end
        % See paper on Utility of Exogenous-Input in Dynamic
Programming
        % Equation 19 for calculation of inner terms for decision
        term1den = a^(2*(N-1-j)*b^2) + term1den;
        if (strcmp(type,'ramp')||strcmp(type,'polynomial'))
            term2 = f(j+1)*(a^(N-1-j))*(H*z(:,j+1))+ term2;
        elseif strcmp(type,'constant')
            term2 = f(j+1)*(a^(N-1-j))*H*z(j+1)+ term2;
        end
%         % Calculation of utility (equation 22 in Utility paper)
%         % Determine inner loop terms for calculation of utility
(equation 22)
        if (strcmp(type,'ramp')||strcmp(type,'polynomial'))
            A2 =   f(j+1)*(a^(N-1-j))*H*z(:,j+1)+ A2;
            Bur = f(j+1)*(a^(N-1-j))*H*z(:,j+1) + Bur;
        elseif strcmp(type,'constant')
            A2 = a^(N-1-j)*H*z(j+1)+ A2;
            Bur = f(j+1)*(a^(N-1-j))*H*z(j+1) + Bur;
        end
    end
    D(k) = -(term1num/(1+term1den))*((a^(N-kz))*x(k)+term2);

    % R term
    Rnum = (a^(N-1-kz))*b;
    R = Rnum/(1+Rden) + R;

    assist = (2*(a^(N-kz))*x(k)*A2)/(1+term1den);
    burden = Bur^2/(1+term1den);
    % Save assistance and burden at every time point
    A(k) = assist;
    B(k) = burden;
    util2(k) = -A(k) - B(k);

    % Clear variables
    term1num = 0;
    term1den = 0;
    term2 = 0;
```

```
    R = 0;
    A2 = 0;
    Bur = 0;
    Rden = 0;
    assist = 0;
    burden = 0;

    % Define transition equation
    if (strcmp(type,'ramp')||strcmp(type,'polynomial'))
        x(k+1) = a*x(k) + b*D(k) + f(k)*H*z(:,k);
    elseif strcmp(type,'constant')
        x(k+1) = a*x(k) + b*D(k) + f(k)*H*z(k);
    end

    % Keep running sum of Decisions for calculation of minimization
    % criteria
    if (k~=1)% Skip if it's the first time
        sumD(k) = D(k).^2 + sumD(k-1);
    else
        sumD(k) = D(k).^2;
    end

end

% Find Optimization Criteria
Jo = x(k+1).^2 + sumD(k);

% Go back and determine J at every sample based on the terminal state
xT = x(N+1);
for k=1:N
    Jrto(k) = xT^2 + D(k)^2;
end
Jrto = [Jrto Jo];

% % Plots
% plot(w,Jrto,'x-g')
return
```

APPENDIX D

DERIVATION OF THE STATE-SPACE MODEL FOR THREE-TYPES OF UNKNOWN EXOGENOUS-INPUTS

"STEPWISE-CONSTANT"-TYPE EXOGENOUS-INPUT

The uncertain exogenous-input is a stepwise constant which can be represented by the following spline-function

$$w(t_i) = w(k) = C_1;\ C_1\text{(k)=unknown stepwise constant; } w_0 = C_1. \qquad \text{(D.1)}$$

To find the lowest order homogenous difference-equation for which the above equation is the general solution, we first take the z-transform of $w(k)$ in (D.1) as follows, assuming, temporarily, that C_1 in (D.1) is uniformly constant

$$W(z) = \mathcal{Z}\{C_1\} = \frac{C_1 \cdot z}{z - 1}. \qquad \text{(D.2)}$$

The denominator of $W(z)$ is the (discrete) characteristic polynomial for the sought homogenous difference-equation. This $w(k)$ satisfies the first-order difference equation

$$zw(k) - w(k) \Rightarrow w(k+1) - w(k) = 0. \qquad \text{(D.3)}$$

To construct the state-space model of $w(k)$, one must first choose a valid set of state-variables for (D.3). In the first-order case (D.3), there will be one state-variable z_1 and the most natural choice for z_1 is

$$z_1 \triangleq w(k)\,. \tag{D.4}$$

Thus, in (D.1), a valid state-evolution model for $w(k)$, with C_1=unknown, stepwise constant is

$$\begin{aligned} &w(k) = z(k); \quad (h = 1.0) \\ &z(k+1) = z(k) + \bar{\sigma}(k)\ ; \quad (g = 1.0), \end{aligned} \tag{D.5}$$

where $\bar{\sigma}$ is a *time-sparse* sequence of Kronecker delta functions with completely unknown arrival times and values. The $\bar{\sigma}$-sequence "cause" the unknown, time-sparse jumps in the value of C_1 in (D.1).

"STEP PLUS RAMP"-TYPE EXOGENOUS-INPUT

The uncertain exogenous-input is a step plus ramp-type of the form

$$w(t_i) = w_k = B_1 + B_2 \cdot k;\ B_{1,2} = \text{unknown stepwise constants};\ w_0 = B_1\,. \tag{D.6}$$

To find the lowest order, homogenous difference-equation for which (D.6) is the general solution, we first take the z-transform of (D.6) as follows, assuming, temporarily, that B_1 and B_2 are uniformly constant

$$W(z) = \mathcal{Z}\{B_1 + B_2 \cdot k\} = \frac{B_1 \cdot z}{z - 1} + \frac{B_2 \cdot z}{(z-1)^2}\,. \tag{D.7}$$

Then, consolidating the right-side of (D.7) into a rational-function of z as follows

$$W(z) = \frac{B_1 \cdot z}{z-1} + \frac{B_2 \cdot z}{(z-1)^2} = \frac{z(B_2 - B_1 + B_1 \cdot z)}{(z-1)^2}. \tag{D.8}$$

The denominator of *W(z)* in (D.8) is the (discrete) characteristic polynomial of the sought homogenous difference-equation. Thus, since

$$(z-1)^2 = z^2 - 2z + 1 = 0, \tag{D.9}$$

it follows that *w(k)* satisfies the 2nd-order, homogenous difference equation

$$w(k+2) - 2w(k+1) + w(k) = 0. \tag{D.10}$$

To construct the state-space model of *w(k)*, one must choose 2-valid state-variables . Thus, if we choose first-order variables such as $z_1 \underline{\underline{\Delta}}\, w(k); z_2 \underline{\underline{\Delta}}\, w(k+1)$, then using (D.10)

$$\begin{bmatrix} z_1(k+1) \\ z_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix}. \tag{D.11}$$

The exogenous-input state-space model for the unknown “step plus ramp” input *w(k)* in (D.6) has the final form

$$\begin{gathered} w(k) = H \cdot z;\; H = \begin{bmatrix} 1 & 0 \end{bmatrix};\; \begin{matrix} z_1(k) \underline{\underline{\Delta}} w(k) \\ z_2(k) \underline{\underline{\Delta}} w(k+1) \end{matrix} \\ \begin{bmatrix} z_1(k+1) \\ z_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} z_1(k) \\ z_2(k) \end{bmatrix} + \begin{bmatrix} \bar{\sigma}_1(k) \\ \bar{\sigma}_2(k) \end{bmatrix};\; G = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}, \end{gathered} \tag{D.12}$$

where the $\overline{\sigma}_i$ are *time-sparse* sequences of Kronecker delta functions with completely unknown arrival times and values and account for the time-sparse jumps in the values of B_1 and B_2 in (D.6).

“CUBIC POLYNOMIAL-SPLINE”-TYPE EXOGENOUS-INPUT

The uncertain exogenous-input is a discrete cubic polynomial-spline-type of the form

$$w(k) = C_1 + C_2 k + C_3 k^2 + C_4 k^3;\ C_i = \text{unknown constant}\ . \tag{D.13}$$

To find the minimal order homogenous difference-equation for which (D.13) is a solution, we first take the z-transform of (D.13) as follows

$$\begin{aligned} W(z) &= \mathcal{Z}\left\{C_1 + C_2 k + C_3 k^2 + C_4 k^3\right\} \\ &= \frac{C_1 z}{(z-1)} + \frac{C_2 z}{(z-1)^2} + \frac{C_3 z(z+1)}{(z-1)^3} + \frac{C_4 z(z^2+4z+1)}{(z-1)^4} \cdot \end{aligned} \tag{D.14}$$

Consolidating the right side of (D.14) into a rational-function of z gives

$$W(z) = \frac{z(C_2 - C_1 - C_3 + C_4 - 3C_1 z^2 + C_1 z^3 + C_2 z^2 + C_3 z^2 + C_4 z^2 + 3C_1 z - 2C_2 z + 4C_4 z)}{(z-1)^4}\ . \tag{D.15}$$

The denominator of $W(z)$ in (D.15) is the (discrete) characteristic polynomial of the sought homogenous difference-equation. Thus, since

$$(z-1)^4 = z^4 - 4z^3 + 6z^2 - 4z + 1\ , \tag{D.16}$$

it follows that w(k) satisfies the fourth-order difference-equation

$$
\begin{aligned}
z^4 w(k) - 4z^3 w(k) + 6z^2 w(k) - 4zw(k) + w(k) = 0 \\
\Rightarrow w(k+4) - 4w(k+3) + 6w(k+2) - 4w(k+1) + w(k) = 0.
\end{aligned}
\tag{D.17}
$$

To construct the state-space model of *w(k)*, we must choose a valid set of (independent) state variables such as

$$
\begin{aligned}
z_1 &\triangleq w(k) \\
z_2 &\triangleq w(k+1) \\
z_3 &\triangleq w(k+2) \\
z_4 &\triangleq w(k+3)
\end{aligned}
. \tag{D.18}
$$

From (D.18) and using (D.17) gives

$$
\begin{bmatrix} z_1(k+1) \\ z_2(k+1) \\ z_3(k+1) \\ z_4(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & -6 & 4 & -1 \end{bmatrix} \begin{bmatrix} z_1(k) \\ z_2(k) \\ z_3(k) \\ z_4(k) \end{bmatrix}. \tag{D.19}
$$

The exogenous-input state-space model for the unknown "cubic polynomial spline" (D.13) has the final form

$$
\begin{aligned}
& w(t) = H \cdot z; H = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}; \\
& \begin{bmatrix} z_1(k+1) \\ z_2(k+1) \\ z_3(k+1) \\ z_4(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & -6 & 4 & -1 \end{bmatrix} \begin{bmatrix} z_1(k) \\ z_2(k) \\ z_3(k) \\ z_4(k) \end{bmatrix} + \begin{bmatrix} \bar{\sigma}_1(k) \\ \bar{\sigma}_2(k) \\ \bar{\sigma}_3(k) \\ \bar{\sigma}_4(k) \end{bmatrix}; G = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & -6 & 4 & -1 \end{bmatrix}.
\end{aligned}
\tag{D.20}
$$

APPENDIX E

DERIVATION OF THE UTILITY DOMAINS FOR THREE-TYPES OF UNKNOWN EXOGENOUS-INPUTS

For all the examples in this appendix, the sequential-decision process is assumed to be governed by the following first-order, linear, state-evolution equation

$$x_{k+1} = ax_k + bD_k + fw_k \, , \tag{E.1}$$

where the discrete integer "stages" k in the sequential-decision problem extend over the time interval $0=t_0 \le k \le 4 = T$ (T =terminal time), and the coefficients (a, b, and f) are scalar and known constants (*time-invariant*).

<u>"STEPWISE-CONSTANT"-TYPE EXOGENOUS-INPUT</u>

The uncertain exogenous-input *w(k)* is a stepwise constant-type of the form

$$w(t_i) = w(k) = C_1; \; C_1(k) = \text{unknown stepwise constant}; \; w_0 = C_1 \, , \tag{E.2}$$

and the corresponding state-space model for the exostate *z(k)* of the exogenous-input *w(k)* is

$$w(k) = hz(k); \quad h = 1; \qquad z(k+1) = gz(k) + \overline{\sigma}(k); \quad g = 1. \tag{E.3}$$

The general utility expression corresponding to (E.1) and (E.3) was derived in Section 5.3 to be

$$\mathscr{U}_{RTO}(k) = -\left[\frac{1}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)} b^2}\right] \times \left[2a^{(N-k)} x_k \sum_{j=k}^{N-1}\left(a^{(N-1-j)} fhz(j)\right) + \left(\sum_{j=k}^{N-1}\left(a^{(N-1-j)} fhz(j)\right)\right)^2\right]. \tag{E.4}$$

To obtain the boundaries separating the positive and negative utility domains in (*x, z*)-space, we set the utility expression in (E.4) equal to zero and solve for the exostate *z(k)* in terms of *x(k).* Thus, we obtain the expression

$$\left(\sum_{j=k}^{N-1}\left(a^{(N-1-j)} fhz(j)\right)\right)\left[2a^{(N-k)} x_k + \left(\sum_{j=k}^{N-1}\left(a^{(N-1-j)} fhz(j)\right)\right)\right] = 0 \tag{E.5}$$

which implies that the first term in parenthesis in (E.5) has to equal zero as well as the second term in the brackets which gives the following

$$z(j) = 0 \text{ and } \left(\sum_{j=k}^{N-1}\left(a^{(N-1-j)} fhz(j)\right)\right) = -2a^{(N-k)} x_k\,. \tag{E.6}$$

The first expression in (E.6) indicates that the utility domain lies in the $z = 0$ plane. The second expression in (E.6) does not readily allow solving for *z*, thus, to simplify visualization of the utility domains, we will look at the second expression in (E.6) at each discrete time *k*. Consequently, beginning at *k*=0, the second expression in (E.6) becomes

$$\left[fhz(0)a^3 + fhz(1)a^2 + fhz(2)a + fhz(3) \right] = -2a^4 x(0). \qquad \text{(E.7)}$$

Next, we apply the RTO-Principle in which we assume that the step of this example will remain constant at C_1 from the beginning of the process at k=0 until the end of the process at k=4. Thus, $z(k)=C_1$ and this gives

$$\left[fhC_1a^3 + fhC_1a^2 + fhC_1a + fhC_1 \right] = -2a^4 x(0). \qquad \text{(E.8)}$$

Setting h=1 as defined in (E.3) and solving for C_1 gives

$$C_1 = -\frac{2a^4 x(0)}{a^3 f + a^2 f + af + f}\text{; at } k\text{=0.} \qquad \text{(E.9)}$$

Following the same procedure at k=1 gives

$$\left[fhz(1)a^2 + fhz(2)a + fhz(3) \right] = -2a^3 x(1). \qquad \text{(E.10)}$$

Next, we substitute $z(k)=C_1$ since by applying the RTO-Principle, we are assuming that the step will remain constant at C_1 from the beginning of the process at k=1 until the end of the process at k=4. This gives

$$\left[fhC_1a^2 + fhC_1a + fhC_1 \right] = -2a^3 x(1). \qquad \text{(E.11)}$$

Once again, setting h=1 and solving for C_1 gives

$$C_1 = -\frac{2a^3 x(1)}{a^2 f + af + f}\text{; at } k\text{=1.} \qquad \text{(E.12)}$$

Following the same steps for k=2 gives

$$C_1 = -\frac{2a^2 x(2)}{af + f}\text{; at } k\text{=2.} \qquad \text{(E.13)}$$

And finally, for k=3, we obtain the following

$$C_1 = -\frac{2ax(3)}{f}\text{; at } k\text{=3.} \tag{E.14}$$

Generalizing equations (E.9), (E.12), (E.13), and (E.14) and setting C_1 back to $z(k)$* gives

$$z(k) = 0 \text{ and } z(k) = 2Kx(k);\; K = -\frac{a^{(N-k)}}{\sum_{j=k}^{N-1}(a^j f)}. \tag{E.15}$$

The above general expression for the utility domains in (x, z)-space of a constant exogenous-input lies between the plane $z(k)$=0 and the intersecting plane defined by the second equation in (E.15).

* Since the utility expressions are in terms of the exostate $z(k)$ at different discrete times k and to simplify the visualization of the utility-domains, the exostate values for $z(0)$, $z(1)$, $z(2)$, and $z(3)$ were set equal to C_1 through the application of the RTO principle. Once the general utility domains were obtained, the notation was reversed to its original condition.

"STEP PLUS RAMP"-TYPE EXOGENOUS-INPUT

The uncertain exogenous-input $w(k)$ is a step plus ramp-type of the form

$$w(t_i) = w_k = B_1 + B_2 \cdot k;\; B_{1,2} = \text{unknown stepwise constants; } w_0 = B_1, \tag{E.16}$$

and the corresponding state-space model for the exostate $z(k)$ of the exogenous-input $w(k)$ is

$$
\begin{aligned}
&w(k) = [1 \quad 0]\begin{pmatrix} z_1(k) \\ z_2(k) \end{pmatrix}; \;\; H = [1 \quad 0] \\
&\begin{pmatrix} z_1(k+1) \\ z_2(k+1) \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}\begin{pmatrix} z_1(k) \\ z_2(k) \end{pmatrix} + \begin{pmatrix} \overline{\sigma}_1(k) \\ \overline{\sigma}_2(k) \end{pmatrix}; \;\; G = \begin{bmatrix} 0 & 1 \\ -1 & 2 \end{bmatrix}.
\end{aligned} \tag{E.17}
$$

The general utility expression corresponding to (E.16) and (E.17) was derived in Section 6.3 to be

$$
\mathscr{U}_{RTO}(k) = -\left[\frac{1}{1+\sum_{j=k}^{N-1} a^{2(N-1-j)}b^2}\right] \times \left[2a^{(N-k)}x_k \sum_{j=k}^{N-1}(a^{(N-1-j)} fH\overrightarrow{z_j}) + \left(\sum_{j=k}^{N-1}(a^{(N-1-j)} fH\overrightarrow{z_j})\right)^2\right],
$$

$$
\tag{E.18}
$$

where due to the second-order exogenous-input exostate $\overrightarrow{z_j} = col.(z_1 \mid z_2)$ of this example, the utility has the slightly modified form of (E.18) where H is a 1x2 matrix.

Following the same steps as in the Stepwise-Constant case, we set the utility expression in (E.18) equal to zero and solve for the exostate *z(k)* in terms of *x(k)*. Thus, we obtain the expression

$$
\left(\sum_{j=k}^{N-1}(a^{(N-1-j)} fH\overrightarrow{z_j})\right)\left[2a^{(N-k)}x_k + \left(\sum_{j=k}^{N-1}(a^{(N-1-j)} fH\overrightarrow{z_j})\right)\right] = 0\,, \tag{E.19}
$$

which implies that the first term in parenthesis in (E.19) has to equal zero as well as the second term in the brackets has which gives the following expressions

$$
\left(\sum_{j=k}^{N-1}(a^{(N-1-j)} fH\overrightarrow{z_j})\right) = 0 \text{ and } \left(\sum_{j=k}^{N-1}(a^{(N-1-j)} fH\overrightarrow{z_j})\right) = -2a^{(N-k)}x_k\,. \tag{E.20}
$$

The first expression in (E.20) indicates that the utility domain lies in the $z = 0$ plane. The second expression in (E.20) does not readily allow solving for z, thus, to simplify visualization of the utility domains, we will look at the second expression in (E.20) at each discrete time k. Consequently, beginning at k=0, the second term in (E.20) becomes

$$a^3 f z_1(0) + a^2 f z_1(1) + af z_1(2) + f z_1(3) = -2a^4 x(0)\,, \tag{E.21}$$

where due to the nature of the matrix H(=[1 0]), the instantaneous utility and utility domains expressions are in terms of only the first exostate variable z_1.

Next we apply the RTO-Principle in which we assume that the Step plus Ramp of this Example will remain at the same value from the beginning of the process at k=0 until the end of the process at k=4. Thus, $z_1(0) = z_1(1) = z_1(2) = z_1(3) = z_1$; solving for z gives

$$\begin{aligned} z_1(k)\left(a^3 f + a^2 f + af + f\right) &= -2a^4 x(0) \\ z_1(0) &= -\frac{2a^4 x(0)}{\left(a^3 f + a^2 f + af + f\right)} \quad \text{; at } k\text{=0.} \end{aligned} \tag{E.22}$$

Discrete stage k=1 gives

$$z_1(1) = -\frac{2a^3 x(1)}{\left(a^2 f + af + f\right)}\text{; at } k\text{=1.} \tag{E.23}$$

Discrete stage k=2 gives

$$z_1(2) = -\frac{2a^2 x(2)}{\left(af + f\right)}\text{; at } k\text{=2.} \tag{E.24}$$

And finally, for k=3, we obtain the following utility domain

$$z_1(3) = -\frac{2a^2 x(3)}{f}; \text{ at } k=3. \qquad \text{(E.25)}$$

Generalizing equations (E.22)-(E.25) gives

$$z_1(k) = 0 \;\text{ and }\; z_1(k) = 2Kx(k);\; K = -\frac{a^{(N-k)}}{\sum_{j=k}^{N-1}(a^j f)}, \qquad \text{(E.26)}$$

where K is a constant that varies with each discrete time and depends on the process state coefficient a and the exogenous-input coefficient f. (E.26) represents the same utility domains that were obtained with the Stepwise Constant. Thus, the above general expression for the utility domains in (x, z)-space of a Step plus Ramp exogenous-input lies between the plane $z(k)=0$ and the intersecting plane defined by the second equation in (E.26).

<u>"CUBIC POLYNOMIAL-SPLINE"-TYPE EXOGENOUS-INPUT</u>

The uncertain discrete-time exogenous-input $w(k)$ is a "cubic polynomial spline" of the form

$$w(k) = C_1 + C_2 k + C_3 k^2 + C_4 k^3;\; C_i = \text{unknown constant}, \qquad \text{(E.27)}$$

and the corresponding discrete state-space model for the exostate $z(k)$ of the exogenous-input $w(k)$ is

$$w(t) = H \cdot z; H = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix};$$

$$\begin{bmatrix} z_1(k+1) \\ z_2(k+1) \\ z_3(k+1) \\ z_4(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & -6 & 4 & -1 \end{bmatrix} \begin{bmatrix} z_1(k) \\ z_2(k) \\ z_3(k) \\ z_4(k) \end{bmatrix} + \begin{bmatrix} \bar{\sigma}_1(k) \\ \bar{\sigma}_2(k) \\ \bar{\sigma}_3(k) \\ \bar{\sigma}_4(k) \end{bmatrix}; G = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 4 & -6 & 4 & -1 \end{bmatrix}. \tag{E.28}$$

The general utility expression corresponding to (E.27) and (E.28) was derived in Section 6.3 and is the same utility expression as for the step plus ramp-type exogenous-input and is as follows

$$\mathscr{U}_{RTO}(k) = -\left[\frac{1}{1 + \sum_{j=k}^{N-1} a^{2(N-1-j)} b^2} \right] \times \left[2a^{(N-k)} x_k \sum_{j=k}^{N-1} (a^{(N-1-j)} fH \overrightarrow{z(j)}) + \left(\sum_{j=k}^{N-1} (a^{(N-1-j)} fH \overrightarrow{z(j)}) \right)^2 \right], \tag{E.29}$$

where due to the fourth-order exogenous-input exostate $\overrightarrow{z(j)} = col.(z_1 \mid z_2 \mid z_3 \mid z_4)$ of this Example, the utility has the slightly modified form of (E.29) but it is still a scalar quantity.

Following the same steps as in the Stepwise-Constant and the Step plus Ramp cases, we set the utility expression in (E.29) equal to zero and solve for the exostate *z(k)* in terms of *x(k)*. Thus, we obtain the expression

$$\left(\sum_{j=k}^{N-1} (a^{(N-1-j)} fH \overrightarrow{z_j}) \right) \left[2a^{(N-k)} x_k + \left(\sum_{j=k}^{N-1} (a^{(N-1-j)} fH \overrightarrow{z_j}) \right) \right] = 0, \tag{E.30}$$

which in turn implies that

$$\left(\sum_{j=k}^{N-1} (a^{(N-1-j)} fH \overrightarrow{z_j}) \right) = 0 \text{ and } \left(\sum_{j=k}^{N-1} (a^{(N-1-j)} fH \overrightarrow{z_j}) \right) = -2a^{(N-k)} x_k. \tag{E.31}$$

The utility domains in (E.31) are exactly the same as the utility domains in (E.20) for the Step Plus Ramp exogenous-input case. Thus, the general utility domains for this example have the form

$$z_1(k) = 0 \ \text{ and } \ z_1(k) = 2Kx(k); \ K = -\frac{a^{(N-k)}}{\sum_{j=k}^{N-1}(a^j f)}. \tag{E.32}$$

where K is a constant that varies with each discrete time and depends on the process state coefficient a and the exogenous-input coefficient f and due to the nature of the matrix ($H = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$), the utility domains in (E.32) are in terms of only the first exostate variable z_1 ($H\overrightarrow{z(k)} = z_1(k)$). (E.32) represents the same utility domains that were obtained with the Stepwise Constant and the Step Plus Ramp exogenous-inputs. Thus, the above general expression for the utility domains in (x, z)-space of a Cubic Polynomial Spline-type exogenous-input lies between the plane $z(k) = 0$ and the intersecting plane defined by the second equation in (E.26).

The utility domains in (E.32) are very general and apply to the N-stage first-order linear SDP of (6.1) and (6.2) when a structured-variation exogenous-input that has the Exostate model of (3.27) is acting on the sequential decision process and the RTO principle has been applied.

REFERENCES

[1] Aoki, M., "Dynamic Programming Approach to a Final-Value Control System with a Random Variable Having an Unknown Distribution Function", *IRE Transactions on Automatic Control,* Vol.5, No.4; September 1960, pp. 270-283.

[2] Aoki, M., *Optimization of Stochastic Systems*, Academic Press, New York, New York, 1967.

[3] Basharin, G.P., Langville, A.N. and V.A. Naumov, “The Life and Work of A.A. Markov”, *Linear Algebra and its Applications*, Vol. 386, Special Issue on the Conference on the Numerical Solution of Markov Chains 2003; July 15, 2004, pp. 3-26.

[4] Bellman, Richard, “An Introduction to the Theory of Dynamic Programming”, The RAND Corporation, Report No. R-245; 1953.

[5] Bellman, Richard, “Some Problems in the Theory of Dynamic Programming”, *The Econometric Society*, Vol. 22, No. 1; January 1954, pp. 37-48.

[6] Bellman, Richard, “Some Applications of the Theory Of Dynamic Programming - A Review”, *Journal of the Operations Research Society of America,* Vol. 2; 1954, pp. 275–288.

[7] Bellman, Richard, “Decision Making in the Face of Uncertainty I”, *Naval Research Logistics Quarterly*, Vol. 1; September 1954, pp. 230-232.

[8] Bellman, Richard, “Dynamic Programming and Multi-Stage Decision Processes Of Stochastic Type”, *Symposium on Linear Programming*, Washington, D. C.; December 1954.

[9] Bellman, R., "A Markovian Decision Process", *Journal of Mathematics and Mechanics*, Vol. 6; 1957, pp. 679-684.

[10] Bellman, R., “Some Aspects of the Mathematical Theory of Control Processes”, the RAND Corporation, Santa Monica, California, Report No. R-313; 1958.

[11] Bellman, R., and Kalaba, R., "Functional Equations in Adaptive Processes and Random Transmission", *IRE Transactions on Information Theory,* Vol.5, No.5; May 1959, pp.271-282.

[12] Bellman, Richard, "The Theory of Dynamic Programming", *Bulletin of the American Mathematical Society*, Vol. 60; 1964, pp. 503–516.

[13] Bellman, R. and S. Dreyfus, *Applied Dynamic Programming*, Princeton University Press, New Jersey; 1964.

[14] Bertsekas, D. P., *Dynamic Programming; Deterministic and Stochastic Models*. Prentice Hall, Englewood Cliffs, New Jersey; 1987.

[15] Bertsekas, D. P., and Tsitsiklis, J. N., "An Analysis of Stochastic Shortest Path Problems", *Mathematics of Operations Research*, Vol. 16; 1991, pp. 580-595.

[16] Bertsekas, D. P., *Dynamic Programming and Optimal Control*. Athena Scientific, Belmont, Massachusetts; 2007.

[17] Blackwell, D., "Discrete Dynamic Programming", *Annals of Mathematics and Statistics,* Vol. 33, No. 2; 1962, pp. 719-726.

[18] Brogan, W. L., "Modern Control Theory", Prentice Hall, Englewood Cliffs, NJ, third edition; 1990.

[19] Brush, S. G., "A History of Random Processes 1. Brownian Movement", In Stud. Hist. Stat. and Prob., ed. M. G. Kendall and R. L. Plackett, 2; London; 1977, pp. 347-382.

[20] Carroll, Y. and C. D. Johnson, "Optimal Sequential-Decisions in the Presence of Exogenous-Inputs with Structured Variations", *Proceedings of the 41st IEEE Southeastern Symposium on System Theory,* The University of Tennessee, Tullahoma, Tennessee; March 15-17, 2009, pp. 300-305.

[21] Carroll, Y. and C. D. Johnson, "Utility of Exogenous-Inputs in Optimal Sequential-Decision Processes", pending submission to *IEEE Journal of Aerospace and Electronic Systems*.

[22] Chen, P., "An Algorithm for Stochastic Control through Dynamic Programming Techniques", *IRE Transactions on Automatic Control,* Vol.7, No.5; October 1962, pp. 110- 119.

[23] Denardo, R. V., "Contraction Mappings in the Theory of Underlying Dynamic Programming", *SIAM*, Vol. 9; 1967, pp. 165-177.

[24] Denardo, E., *Dynamic Programming: Models and Applications*. Dover Publications, Mineola, New York; 2003.

[25] Derman, C., "On Sequential Decisions and Markov Chains", *Management Science,* Vol. 9, No. 1; October 1962, pp. 16-24.

[26] Fox, B. L., "Semi-Markov Processes: A Primer", P-3577-1, The RAND Corporation, Santa Monica, California; 1967.

[27] Grinold, R. C., "A Generalized Discrete Dynamic Programming Model", *Management Science,* Vol. 20, No. 7, Theory Series; March 1974, pp. 1092-1103.

[28] Howard, R. A., *Dynamic Programming and Markov Processes.* The M.I.T. Press, Cambridge, Massachusetts; 1960.

[29] Jaquette, S. C., "A Utility Criterion for Markov Decision Processes", *Management Science*, Vol. 23, No. 1; 1976, pp. 43-49.

[30] Johnson, C. D., "Optimal Control of the Linear Regulator with Constant Disturbances", *IEEE Transactions on Automatic Control,* Vol. AC-13; August 1968, pp. 416-421.

[31] Johnson, C. D., "Further Study of the Linear Regulator with Disturbances: The Case of Vector Disturbances Satisfying a Linear Differential Equation", *IEEE Transactions on Automatic Control,* Vol. AC-15; April 1970, pp. 222-228.

[32] Johnson, C. D., “Accommodation of Disturbances in Linear Regulator and Servo-Mechanism Problems”, *IEEE Transactions on Automatic Control* (Special Issue on the “Linear-Quadratic-Gaussian Problem”), Vol. AC-16, No. 6; December 1971, pp. 635-644.

[33] Johnson, C. D., “Algebraic Solution of the Servomechanism Problem with External Disturbances”, *Journal of Dynamic Systems, Measurement, and Control*, Vol. 96, Ser. G; March 1974, pp. 25-35.

[34] Johnson, C. D., “Discrete-Time Disturbance-Accommodating Control with Applications to Missile Digital Control”, *AIAA Journal of Guidance and Control*, Vol. 4, No. 2; March-April 1981, p. 116.

[35] Johnson, C. D., “A Discrete-Time Disturbance Accommodating Control Theory for Digital Control of Dynamic Systems”, Chapter 6 in the book: *Control and Dynamic Systems*. Vol.18, edited by C. T. Leondes, Academic Press; 1982, pp. 223-315.

[36] Johnson, C. D., “A General Theory of “Discrete/Continuous” Type Discrete-Time Control for Linear Dynamical Systems”, *Proceedings of the 30th IEEE Southeastern Symposium on System Theory,* West Virginia University, Morgantown, West Virginia; March 8-10, 1998, pp. 56-61.

[37] Johnson, C. D. and B. Morgan, “Some Little-Known Results of Bob Bass in Mathematical System Theory”, *Proceedings of the 1998 American Control Conference*; Philadelphia, Pennsylvania; June 1998, p. 424,

[38] Johnson, C. D., “Beyond Bellman’s Principle of Optimality: The Principle of Real-Time Optimality”, *Proceeding of the 37th IEEE Southeastern Symposium on System Theory*, March 2005, pp. 326-335; IEEE#05EX961.

[39] Johnson, C. D., “Lectures on Disturbance-Accommodating Control”, Course EE-703, University of Alabama in Huntsville, Huntsville, Alabama; circa 2005.

[40] Kalman, R., and Koepcke, R., “Optimal Synthesis of Linear Sampling Control Systems Using Generalised Performance Indices”, *Transactions of the American Society of Mechanical Engineers,* Vol. 80*;* 1958, p. 1820.

[41] Kalman, R. E., “Contributions to the Theory of Optimal Control”, Bol, Soc. Mat. Mexicana, Vol. 5; 1960, pp. 102.

[42] Kalman, R. and R. Bucy, “New Results in Linear Filtering and Prediction", *Journal of Basic Engineering*, ASME, Vol. 83D; 1961, pp. 95-108.

[43] Kelly, W. C., “Homing Missile Guidance with Disturbance-Utilizing Control”, *Proceedings of the 12th IEEE Southeastern Symposium on System Theory,* Virginia Beach, Virginia; May 1980, pp. 260-268.

[44] Kirk, Donald E., "An Introduction to Dynamic Programming", *IEEE Transactions on Education,* Vol.10, No.4; December 1967, pp.212-219.

[45] Larson, R. E. and Robert P. Hostetler, *Precalculus, Second Edition*. Heath and Company, Lexington, Massachusetts; 1989.

[46] Meyn, S. P. and R.L. Tweedie, *Markov Chains and Stochastic Stability*. Springer-Verlag, London; 1993.

[47] Nemhauser, G., *Introduction to Dynamic Programming*, John Wiley and Sons, New York; 1966.

[48] Puterman, M., *Markov Decision Processes*, John Wiley and Sons, New York; 1994.

[49] Ruszczyński, A., "Risk-Averse Dynamic Programming For Markov Decision Processes", *Mathematical Programming*, Series B 125; 2010, pp. 235-261.

[50] Tchamran, A., "An Introduction to a Class of Optimal Control Problems", *IEEE Transactions on Education,* Vol.9, No.4; December 1966, pp.215-219.

[51] Veinott, A. Jr., "Discrete Dynamic Programming with Sensitive Discount Optimality Criteria", *Annals of Mathematical Statistics*, Vol. 40; 1969, pp. 1635-1660.