

Applications of adjoint theory to problems in aerospace and defence science

A. Rajagopalan¹ D. Bucco²

(Received 12 August 2010; revised 3 September 2010)

Abstract

The adjoint simulation method is an efficient, computerised method for the performance analysis of linear time varying systems excited by deterministic and/or stochastic inputs. It is based on the impulse response function (or weighting function) of the so-called adjoint system—an associated linear system derived from the original system by clever utilisation of the mathematical principle of duality. We demonstrate the power of adjoint simulation in the context of guided missile homing loop studies and present a Matlab tool for its speedy implementation and execution. As an example, the tool analyses the miss distance performance of a generic guided missile against an evading target in two dimensional space.

Contents

1 Introduction

C698

<http://anziamj.austms.org.au/ojs/index.php/ANZIAMJ/article/view/2424>

gives this article, © Austral. Mathematical Soc. 2010. Published September 23, 2010. ISSN 1446-8735. (Print two pages per sheet of paper.) Copies of this article must not be made otherwise available on the internet; instead link directly to this URL for this article.

<i>1 Introduction</i>	C698
2 Adjoint method	C699
2.1 Deterministic inputs	C700
2.2 Stochastic inputs	C702
3 Adjoint construction rules	C703
4 Adjoint software tool	C704
4.1 COVAD toolbox overview	C704
4.2 Missile application	C707
5 Conclusion	C713
6 Acknowledgement	C713
References	C713

1 Introduction

Systems analysts and engineers are often confronted with understanding the behaviour and performance of complex dynamic systems driven by various inputs predominantly of a stochastic nature. Some typical applications include miss distance studies of guided missiles, fire control problems, navigation problems and analysis of circuits due to random noise. For these systems, the most general method of tackling the problem and gaining critical statistical insight into the nature of the response is via the use of the Monte Carlo simulation technique [1]. However, when the system is linear or can be suitably linearised, then invoking the Monte Carlo method would be excessive and unwarranted. Instead, the stochastically driven linear system may be more economically and rapidly studied using better matched linear techniques. One such technique is based on adjoint theory [2, 3] and is traditionally known as the adjoint simulation method (or modified adjoint method).

The adjoint simulation method is a computerised method for the performance

analysis of linear time varying (LTV) systems. Mathematically, it is derived from the formal theory of linear differential operators and their associated adjoints. Traditionally, the technique has been very popular in the study of guided missile homing loops and, in general, has been employed during the preliminary and conceptual definition stages of new missile programs. For this work, the LTV system under study is represented in the form of a simulation block diagram. The associated adjoint system is also represented in block diagram form. The adjoint system is constructed by manual manipulation of the original system block diagram in accordance with a given set of rules. These rules were first proposed by Laning and Battin [4] as part of their adjoint simulation work on analogue computers.

In general, given an LTV system with n inputs and m outputs, the adjoint method determines the sensitivity of any of the outputs at a fixed time to each of the n inputs. If the inputs are stochastic in nature, then this approach can be formulated to compute the Root Mean Squared (RMS) value of the system states of interest without resorting to ensemble computations as generally required by the Monte Carlo method [1]. Hence, the technique provides the analyst with a simple but powerful alternative to the Monte Carlo approach in those cases when linearisation is justified.

The purpose of this paper is to describe the theory of the adjoint simulation method as applied to LTV systems, to present a pragmatic implementation of the theory using block diagram representation, and to outline the development of a software package designed to automate the adjoint system construction process within the MATLAB/SIMULINK environment [5]. An example drawn from the open literature illustrates the solution process to a typical problem encountered in aerospace and defence science.

2 Adjoint method

We describe two cases: deterministic and stochastic.

2.1 Deterministic inputs

Consider an LTV system described by the following ordinary differential equation

$$\frac{d^n \mathbf{y}}{dt^n} + \mathbf{a}_{n-1}(t) \frac{d^{n-1} \mathbf{y}}{dt^{n-1}} + \cdots + \mathbf{a}_1(t) \frac{d\mathbf{y}}{dt} + \mathbf{a}_0(t) \mathbf{y} = \mathbf{u}, \quad (1)$$

where \mathbf{y} denotes the system output, \mathbf{u} is its deterministic input and the coefficients are functions of time.

According to linear systems theory [6], the solution to the above differential equation is

$$\mathbf{y}(t) = \int_{t_i}^t \mathbf{u}(\tau) \mathbf{w}(t, \tau) d\tau, \quad (2)$$

where t_i is the time at which the input is applied while $\mathbf{w}(t, \tau)$ denotes the system weighting function or impulse response function. Physically, the function $\mathbf{w}(t, \tau)$ represents the response of the system at time t to a unit impulse applied at the input at time τ . While equation (2) gives the formal solution to the problem under consideration, its practical usefulness is limited. The difficulty is that the variable of integration corresponds to the impulse application time. Thus, in order to generate $\mathbf{w}(t, \tau)$ required for the above solution, it becomes necessary to simulate the system response for each impulse application time. Once the weighting function is obtained in this manner, it is then necessary to carry out the integration in equation (2) numerically.

Since every linear time varying system has a corresponding adjoint system, a better and more efficient alternative is to reformulate equation (2) in terms of the weighting function of the associated adjoint system. To this end, utilising the so-called modified adjoint technique [6] has the effect of requiring only one computer simulation run to yield the desired weighting function. Now, if the adjoint system is subjected to an impulsive input at time τ^* , then the resulting system response at observation time t^* , known as the adjoint

weighting function, is denoted by $w^*(t^*, \tau^*)$. A fundamental relationship exists between a linear time varying system and its corresponding adjoint system [2], namely

$$w^*(t_F - t_1, t_F - t_2) = w(t_2, t_1), \quad (3)$$

where t_1 and t_2 are the impulse application and observation times, respectively, of the original system and t_F denotes final time. Thus, on substitution of (3) into (2) and, after a change of variable, we get

$$y(t) = \int_{t_F-t}^{t_F-t_i} u(\tau - \eta) w^*(\eta, t_F - t) d\eta. \quad (4)$$

Here, the variable of integration appears as the first argument in the adjoint weighting function and thus the solution for the output can be computed with just one computer simulation run. For example, if the input to the system is a step of magnitude K which is applied at time zero, and we seek the solution at final time t_F , then the integral (4) becomes

$$y(t_F) = K \int_0^{t_F} w^*(\eta, 0) d\eta. \quad (5)$$

Consequently, the original system response at the final time can now be computed in one simulation run by integrating the weighting function of the corresponding adjoint system.

This procedure extends to include many step input disturbances as well as other types of deterministic inputs such as ramp or sinusoidal inputs. Furthermore, since the system is linear, one adjoint computer simulation yields the system response to a combination of disturbances, along with a detailed error budget showing how each disturbance influences total system performance.

2.2 Stochastic inputs

Suppose the linear system is driven by some noise process $\mathbf{n}(t)$. The system output is now random in nature:

$$\mathbf{y}(t) = \int_{t_i}^t \mathbf{n}(\tau) \mathbf{w}(t, \tau) d\tau. \quad (6)$$

Squaring both sides of (6) and taking expectations yields the mean square value of the output, namely

$$\mathbb{E}[\mathbf{y}^2(t)] = \int_{t_i}^t \phi(\tau) \mathbf{w}^2(t, \tau) d\tau, \quad (7)$$

where we assume the noise input to be non-stationary white noise with autocorrelation function $\mathbf{R}(t, \tau) = \phi(t) \delta(t - \tau)$. For stationary white noise input, the output mean square value reduces to

$$\mathbb{E}[\mathbf{y}^2(t)] = \phi_0 \int_{t_i}^t \mathbf{w}^2(t, \tau) d\tau, \quad (8)$$

where the white noise power spectral density $\phi(t) = \phi_0$ is double sided with dimension unit^2/Hz . Again, the problem with computation of the weighting function is circumvented by considering the corresponding adjoint system. In this case,

$$\mathbb{E}[\mathbf{y}^2(t)] = \phi_0 \int_{t_f-t}^{t_f-t_i} [\mathbf{w}^*(\eta, t_f - t)]^2 d\eta. \quad (9)$$

If the noise input begins at $t_i = 0$ and the observation time is taken as the final time t_f , the mean square response simplifies to

$$\mathbb{E}[\mathbf{y}^2(t_f)] = \phi_0 \int_0^{t_f} [\mathbf{w}^*(\eta, 0)]^2 d\eta. \quad (10)$$

Thus the mean square response of the original system due to a white noise input is determined by computing the weighting function of the adjoint

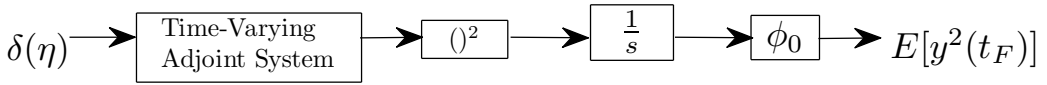


FIGURE 1: Adjoint solution for white noise input.

system in one computer simulation run. This is conveniently depicted in block diagram form in Figure 1. Because of the superposition principle, the above procedure extends to the case of many white noise inputs. In this case, one adjoint simulation run yields an exact statistical solution of the noise-driven system including an error budget showing how each white noise error source contributes to the total system response. Coloured noise inputs can be treated in a similar fashion by the use of appropriate shaping filters.

3 Adjoint construction rules

From the above theory, for linear time varying systems, the adjoint simulation method offers tremendous efficiencies over other methods as a performance projection tool. This had been realised early on by many weapons analysts and cleverly utilised in the preliminary design stages of many current missile systems.

One of the difficulties with this approach is the construction of the adjoint system from which the desired weighting function must be computed. Although several methods for the construction process have been published, by far the most popular method has been that proposed by Laning and Battin [4]. This method works directly on the original simulation block diagram by application of a set of rules. The adjoint construction rules and their practical implementation using block diagram algebra are covered in the book by Zarchan [2].

4 Adjoint software tool

For very complex systems with many feedback paths, application of the adjoint rules can be extremely tedious, time consuming and error prone. If the original simulation block diagram is implemented in SIMULINK, then the process required to construct the adjoint system can be conveniently automated to minimise these shortcomings. A tool has been developed to achieve this. This tool, which forms part of a software package known as COVAD [5], is described below.

4.1 COVAD toolbox overview

The COVAD toolbox has been designed to enable an analyst to easily perform a range of analyses such as adjoint analysis on mathematical models of Multi-Input Multi-Output (MIMO) dynamic systems. It is built employing the widely used SIMULINK modelling tool that is part of the MATLAB suite of graphical modelling products. When looking at the advantages of using SIMULINK for modelling and simulation, in contrast with other approaches, consult the white-paper by Moretti [7]. The COVAD toolbox ties together the following aspects.

- MATLAB scripts, which provide the ability to leverage off the abundant and well documented MATLAB Application Programming Interface (API). They have been used to code up the Graphical User Interfaces (GUIs) needed by the user to interact with the toolbox such as is shown in Figures 2–3, as well as to automate the process of producing the adjoint variant for a given forward time SIMULINK model.
- The SIMULINK modelling environment, which has a huge array of pre-built library blocks as well as an API of its own, includes options to customise the block interfaces. A custom library has been created using this resource to put together the generic blocks used by models which

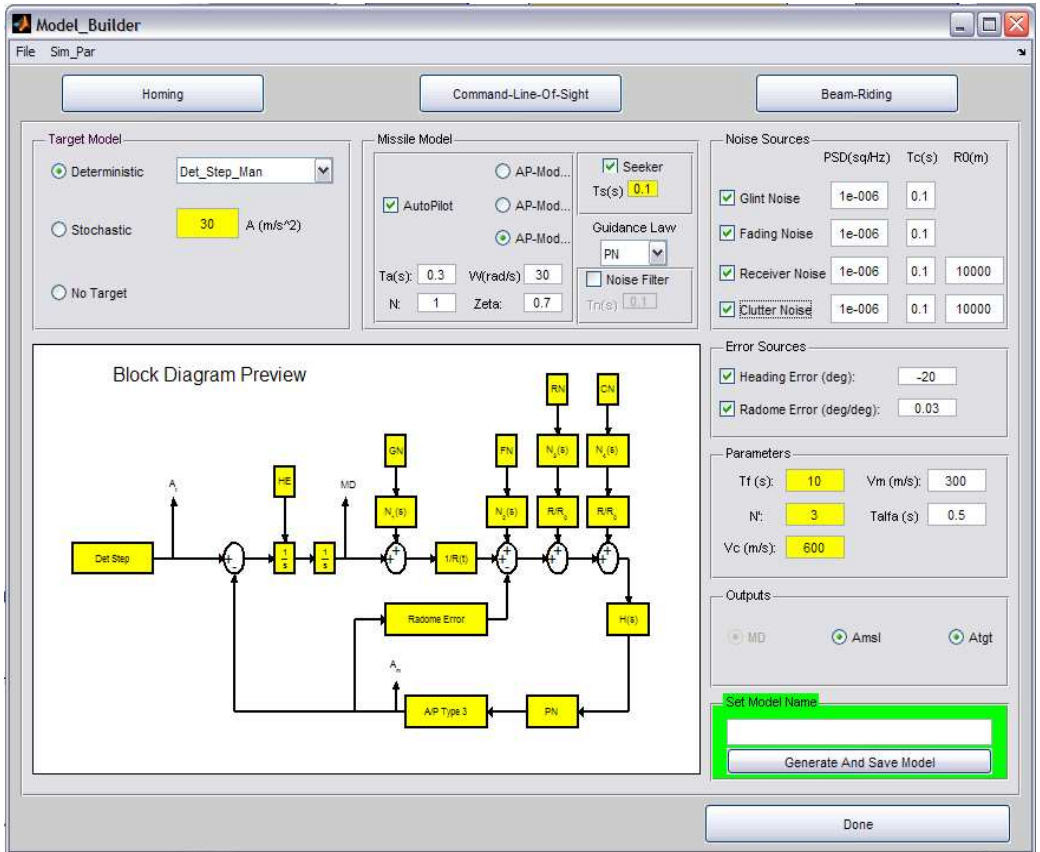


FIGURE 2: first COVAD software screenshot.

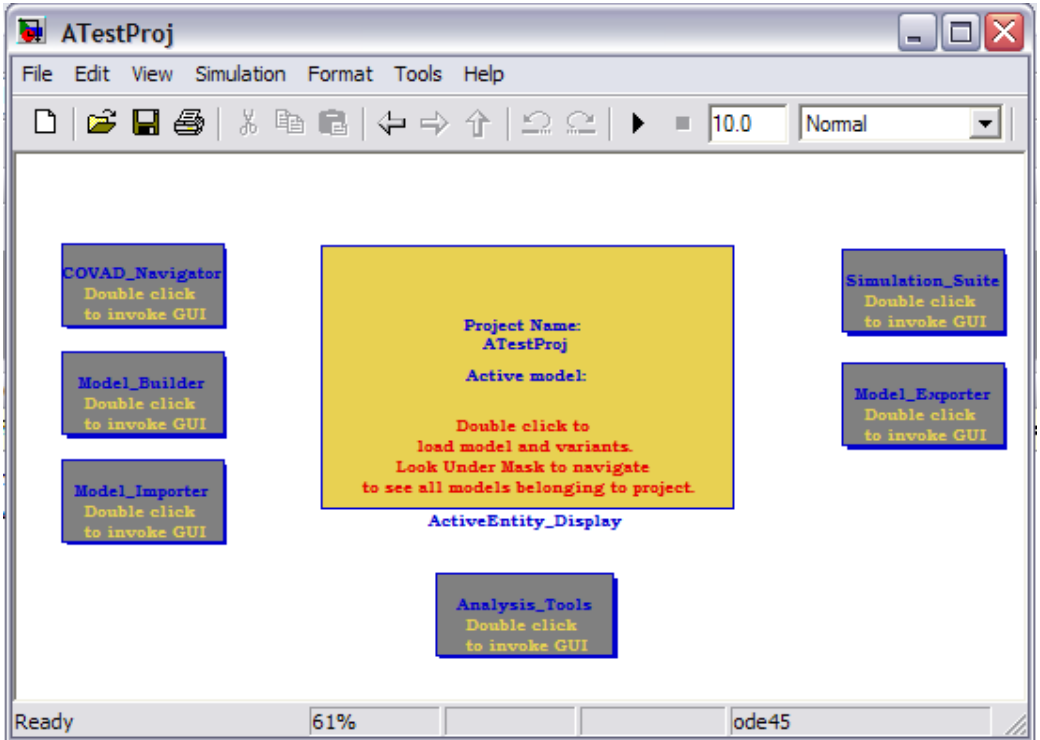


FIGURE 3: second COVAD software screenshot.

lend themselves to adjoint analysis and are part of the missile homing loop domain. This allows for effective reuse.

- MAT files which allow for exporting of simulation results data to other domains, such as excel, as well as allowing the retrieval and display of data by existing Matlab plotting functions.

The program is built with the intention of enabling users to group together SIMULINK models and their variants into project directories which can be placed anywhere in the directory structure. This is analogous to the workflow used in many modern integrated development environments (IDEs). There is also the ability to add new analysis capabilities via new scripts without disturbing the existing code. This is due to the modular software architecture philosophy followed throughout the development. To summarise, this involves

- grouping related functions in separate scripts,
- using information sharing between application components via supported and hence robust data structures, and
- utilising native SIMULINK and MATLAB API capability as much as possible when writing functions.

4.2 Missile application

Consider the planar missile-target engagement geometry depicted as the top diagram of Figure 4. Here we have a radio frequency (RF) guided missile intercepting an incoming target. Both missile and target speed are assumed constant. Initially, both missile and target are on a collision course. The governing equations describing small perturbations about the collision course were given by Zarchan [2]. The bottom diagram of Figure 4 represents the equations in block diagram form. For homing guidance, the missile relies on its RF seeker to track the target. A model of the seeker is approximated by a transfer function as shown in the figure. The signals from the seeker are then

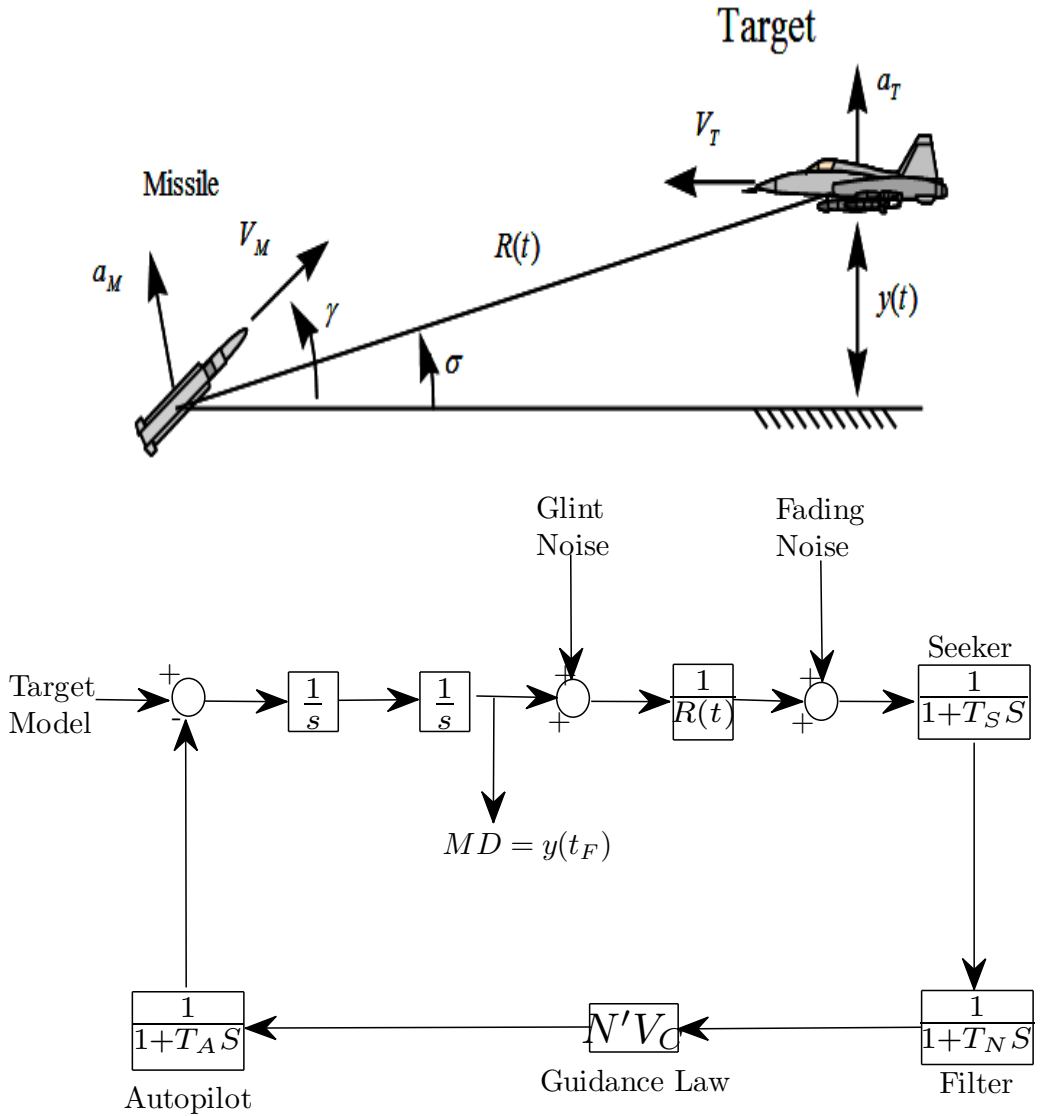


FIGURE 4: Missile and target engagement geometry and associated homing loop.

filtered and used by the guidance computer to generate the guidance commands for the autopilot. The autopilot stabilises the missile and deflects the fins to cause the desired manoeuvre. The missile then reacts aerodynamically and alters the flight path.

In this example, the filter and autopilot/aerodynamic response of the missile are modelled using appropriate transfer functions. The guidance law employed is the traditional proportional navigation guidance law and has the form $\mathbf{a}_C = N'V_C\dot{\sigma}$, where N' is the effective navigation ratio, V_C is the closing velocity, $\dot{\sigma}$ is the rate of change of sight line angle and \mathbf{a}_C denotes the missile commanded acceleration. Radar noise, which contaminates the seeker's measurement of the line of sight to the target, is added in the model in the form of glint and receiver (fading) noise. These are modelled as zero mean white noise processes.

The target model for this example is developed as follows. Prior to intercept, the target is assumed to execute, at random, an evasive step manoeuvre of magnitude A . The random time T for the target manoeuvre is uniformly distributed between zero and the final homing time t_F with probability density function

$$p_t(t) = \begin{cases} \frac{1}{t_F}, & 0 < t \leq t_F, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

Zarchan [2] showed that the above target model can be equivalently represented by a linear shaping filter driven by white noise. This model has the same statistical characteristics, up to second order, as that based on the probability density function given by (11). The power spectral density of the zero mean, input white noise process is $\phi_n = A^2/t_F$. Table 1 summarizes the parameter values used in this example. After inclusion of the target model into the homing loop block diagram of Figure 4 and taking into account the parameter values given in Table 1, we obtain the SIMULINK implementation given as top diagram of Figure 5. The adjoint toolbox COVAD now automatically converts this SIMULINK model into an associated SIMULINK adjoint system for further analysis. The SIMULINK adjoint system is shown as bottom

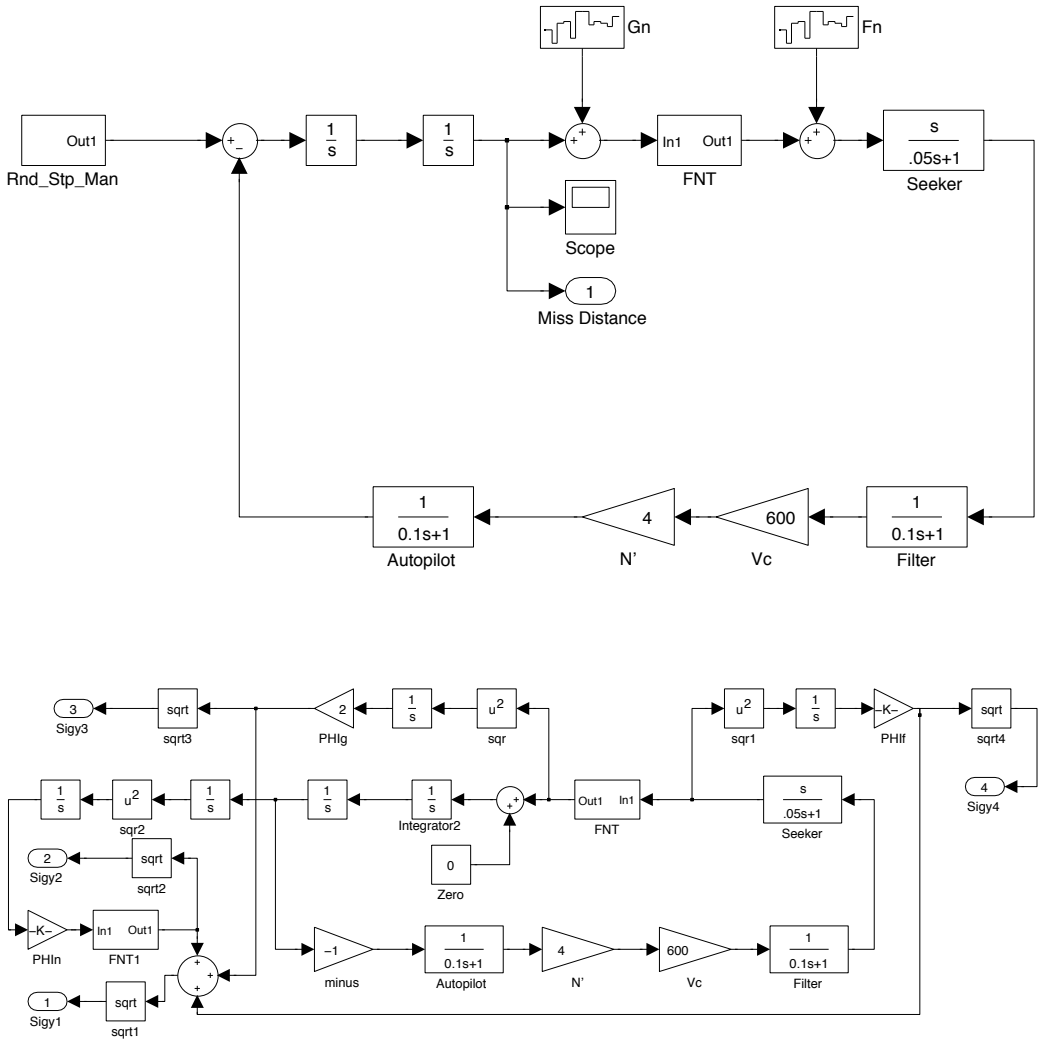


FIGURE 5: Simulink system and corresponding adjoint.

TABLE 1:

Parameter	Nominal Value
Time of Flight, t_F	5 s
Seeker Time Const, T_S	0.05 s
Filter Time Const, T_N	0.1 s
Autopilot Time Const, T_A	0.1 s
Tgt-Man-Mag, A	3g
Glint Noise PSD, ϕ_G	2 m ² /Hz
Fading Noise PSD, ϕ_F	10 ⁻⁶ rad ² /Hz
PN Ratio, N'	4

diagram of Figure 5. Note that the impulse response of the adjoint system has been computed by imposing an initial condition of unity on `integrator2` in the diagram. Standard SIMULINK tools further analyse the system. The top plot of Figure 6 compares the results from the adjoint simulation with those generated using a Monte Carlo method. The results give the RMS miss distance of the engagement as a function of flight time. The adjoint results were obtained in one simulation run of the Simulink system. The Monte Carlo results are based on the physical model of the target manoeuvre (as opposed to the shaping filter approach required for the adjoint model) and required 200 runs for each flight time t_F considered. In addition to the total RMS miss distance results, the adjoint solution also provides an error budget consisting of the contributions that each input has on total system performance. The error budget information is displayed as the bottom plot in Figure 6. The error budget indicates that, in this example, glint noise is the biggest contributor to miss distance.

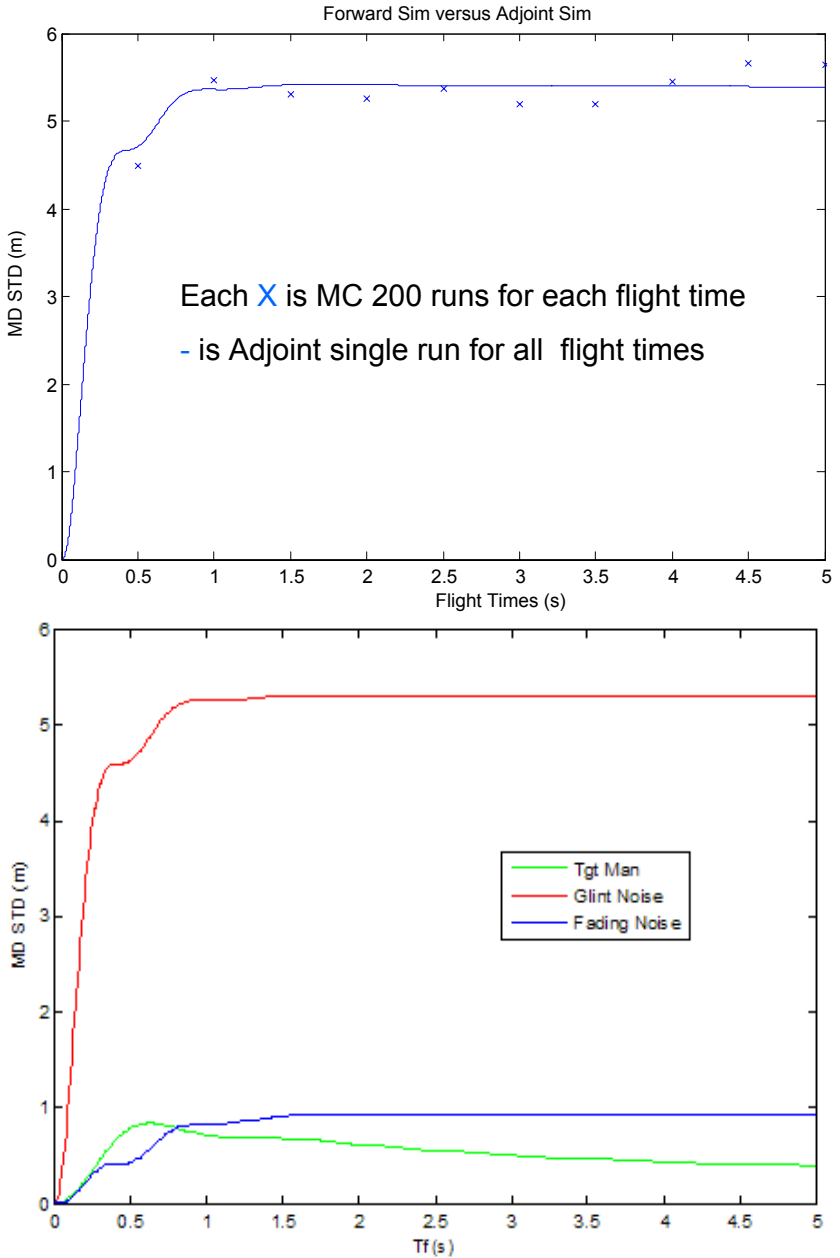


FIGURE 6: simulation results.

5 Conclusion

Mathematicians have long paid attention to the adjoint operator and duality theorems in their studies of the existence of solutions to differential equations. The adjoint simulation method has been employed here to investigate the response and sensitivity of linear time varying systems to stochastic inputs. MATLAB/SIMULINK software has been developed to facilitate applications of the method to problems encountered in aerospace and defence science.

6 Acknowledgement

The authors are grateful to Dr Martin Weiss of TNO, The Netherlands, for his constructive comments and advice on aspects of the adjoint method and valuable suggestions for further developing the adjoint tools in MATLAB.

References

- [1] Statistical performance analysis of nonlinear stochastic systems by the Monte Carlo method volume 23-1 of *Mathematics and Computers in Simulation J.*, pages 21–23, 1981. doi:10.1016/0378-4754(81)90004-5 C698, C699
- [2] Zarchan, P. *Tactical and Strategic Missile Guidance*, 5th Edition, AIAA Progress in Astronautics and Aeronautics, 2007. C698, C701, C703, C707, C709
- [3] Weiss, M. Adjoint Method for Missile Performance Analysis on State-Space Models, volume 28-2 of *Guidance, Control and Dynamics J.*, pages 236–248, AIAA,2005. C698

- [4] Laning, J. H. & Battin, R. H. *Random Processes in Automatic Control*, McGraw-Hill Book Company, New York, NY, 1956. C699, C703
- [5] D. Bucco & M. Weiss. Development of a Matlab/Simulink Tool to Facilitate System Analysis and Simulation via the Adjoint and Covariance Methods, *AIAA MST Conference*, Aug 2007.
<http://www.aiaa.org/agenda.cfm?lumeetingid=1501&formatview=1&dateget=22-Aug-07> C699, C704
- [6] DeRusso, P. M., Roy, R. J., Close, C. M., & Desrochers, A. A. *State Variables for Engineers, 2nd Edition*, John Wiley & Sons, Inc., New York, NY, 1998. C700
- [7] Moretti, G. Best Practices for Adopting Model-Based Design in Electronic System Development, White-paper, *Gabe on EDA*, 2007.
<http://www.mathworks.com/products/simulink/technicalliterature.html> C704

Author addresses

1. **A. Rajagopalan**, Weapons Systems Division, 1.G.30, P.O. Box 1500, DSTO Edinburgh, South Australia 5111, AUSTRALIA.
<mailto:arvind.rajagopalan@dsto.defence.gov.au>
2. **D. Bucco**, Weapons Systems Division, 1.C.42, P.O. Box 1500, DSTO Edinburgh South Australia 5111, AUSTRALIA.
<mailto:domenic.bucco@dsto.defence.gov.au>