Application of Formal Verification Methods to the Analysis of Bearings-only Ballistic Missile Interception Algorithms

M. Moulin*, E. Bendersky **, and L. Gluhovsky⋆
* Ph.D., research staff member
** student research staff member
⋆ research staff member
IBM Haifa Research Laboratories,
Haifa 31905, Israel

ABSTRACT

The paper introduces the application of a formal verification (model checking) technique for the improvements in performance of antimissile interception. In particular, we look into the problem of tracking a nonmaneuvering ballistic missile in its boost phase, based on bearings-only measurements. The interception parameter estimations are essential for the implementation of efficient guidance laws. Assuming knowledge of a target model, a standard Extended Kalman Filter (EKF) was applied. This filter is known to produce biased estimations of the range and range rate. The estimation debiasing procedure consists in addition of an artificial pseudo-noise in order to compensate for the errors in the state predicition by using a larger modified process noise covariance in the covariance prediction equation. The value of the pseudo-noise covariance matrix was obtained by using the RuleBase formal verification engine - a formal verification tool developed by the IBM Haifa Research Laboratory. We also checked the consistency properties of the augmeneted proportional navigation guidance law. Numerical results of a realistic representative case are presented.

1 Introduction to the Interception Problem

This work studies the impact of formal methods (model checking) [1] on the problem of moving target estimation. This problem, also called the interception problem, comprises a highly nonlinear model of target and pursuer motion, with incomplete state measurements. Traditionally, the only available data are bearing angle measurements, which are corrupted by Gaussian noise.

Various types of stochastic filters have been employed to estimate target motion. The nonlinear tracking process employs an Extended Kalman Filter (EKF), or a multi-model

adaptive Kalman filter [2], [3]. These filters perform state estimation in the Cartesian coordinate frame, and update the system in the spherical coordinates, thereby exploiting the linearity in both coordinate systems. The filters determine the system's present state by integrating estimation results with measurement noise parameters and system model knowledge. The drawback of this technique is the bias in the EKF estimations.

The EKF estimation of the interception state is based on the target acceleration modeling or computation. The information concerning the target acceleration impact is also essential for implementation of the guidance laws. In order to guarantee interception, modern guidance laws contain a term representing target acceleration [4], in addition to the classical proportional navigation term.

Target acceleration can be considered, essentially, as a white noise process. The standard state estimation techniques, such as EKF, have serious application faults due to the assumption of uncorrelated accelerations. The more advanced correlated model-based filter is unable to track target motion when the acceleration changes in a nearly discontinuous manner. Some improvement can be achieved by employing the white or correlated Gaussian process, with randomly switching means, as a relevant target acceleration model. Another approach to target acceleration modeling is the analysis of its relationship with the aerodynamic characteristic of the target. For example, a ballistic missile in its boost phase employs a simple target acceleration model, with the realistic assumption that the target accelerates with constant thrust [4].

Generally, the closed-loop tracking system is comprised of a continuous-time engagement model, a continuous-time guidance law, a discrete state estimator (EKF), and a target acceleration model. This system evolves in continuous time with discrete jumps at particular time instances. The tuning of this hybrid system tends to become tedious. Techniques developed for purely continuous, or purely discrete, systems are not directly applicable. The direct discretization of the continuous subsystem is undesirable because of the inherent drawbacks of the discretization process. Still, the internal quasi-discretization operation of digital computer simulation tools, such as Matlab/Simulink, is unavoidable. Using these tools, the control system is designed and then simulated hundreds of times in the Monte Carlo mode. Note that full coverage of the system behavior cannot be achieved in simulation mode.

Formal methods [1] can significantly relieve the computational burden of the system verification process, using a traditional Matlab-based discrete representation of the system. In contrast to simulation, formal methods provide full coverage of all possible cases (scenarios) by verifying a logical model, in order to satisfy/unsatisfy the specified properties. These properties have such forms as "the given condition holds from now on in all cases (i.e., transitions from one finite state machine state to another), beginning in the current state (i.e., initial conditions)," or "the condition holds now, or will hold in the future, at least once beginning in the current state." Such properties can be successfully applied to describe the complete behavior of the system.

2 Formal Methods or Model Checking in Brief

Model checking is a well-established completely automatic approach to verification which analyzes a system against desired properties [1]. Model checking has been used extensively in hardware design and is currently being used to analyze software systems. Model checking is a technique that relies on building a finite model of a system and checking that a desired property holds true in that model. In this approach, the properties are expressed in temporal logic [1] and systems are modeled as finite state transition systems. An efficient search procedure is used to check whether a given finite state transition system is a viable model for the real system. The check is performed as an exhaustive state space search, which is guaranteed to terminate, since the model is finite. The technical challenge in model checking is in devising algorithms and data structures that can handle large search spaces. A data structure called ordered binary decision diagrams (BDDs) is often used to efficiently represent state transition systems, thereby increasing the size of the systems that can be verified. Model checking provides useful information about a system's correctness, as well as counterexamples, which usually represent subtle errors in design, and can thus be used to aid in debugging.

The Bounded Model Checking approach (BMC) is a recent, rapidly-developing area of model checking, in which a specific Boolean formula is constructed from the system under test. This formula can be satisfied if and only if the underlying state transition system can realize a finite k sequence of state transitions that reaches certain states of interest. If such a path segment cannot be found at a given length k, the search can be continued for a larger k. Note that when a check is done for a specific path segment of length k, all path segments of length k are examined. The Boolean formula formed is given to a satisfiability solving program [5] and, if a satisfying assignment for the formula is found, that assignment represents a witness for the path segment of interest. Unlike BDD-based approaches, bounded model checking tools do not require exponential space, and large systems can be checked very quickly, since the state space is searched in an on-the-fly, heuristically-learned, order. Still, the applied method is generally incomplete, which means that one cannot be guaranteed a true or false determination for every property. This is because the length of the propositional formula subject to satisfiability solving grows with each time step and greatly inhibits the capacity of finding long witnesses (or counterexamples) for a large k, and the ability to automatically check all possible paths.

In formal verification, the properties of the system are formulated in a formal specification language [6]-[8]. These properties are typically translated into standard temporal logic, possibly augmented with auxiliary state machines. Properties describe the relationships between Boolean expressions over time. For example,

$$\textbf{always}(\textit{request} \rightarrow \textbf{next}\ \textit{acknowledge}) \tag{1}$$

is a temporal property stating that whenever (**always**) the signal *request* is asserted, then ($\rightarrow$) at the next cycle (**next**), the signal *acknowledge* is asserted.

Today, model checkers are routinely expected to handle systems with thousands of state variables. As a result, model checking is now powerful enough that it is becoming widely used in industry to aid in the verification of newly developed designs [6] - [8].

3 Illustrative Realistic Interception Model

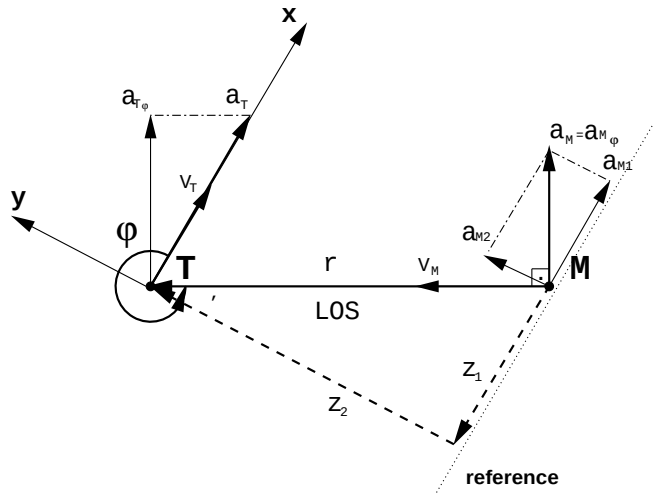

Figure 1: Geometry of the planar tracking problem

The geometry of the illustrative example of the nonmaneuvering target pursuit is depicted in Figure 1. Both the target **T** and pursuing missile **M** are assumed to be point masses moving in a plane. The polar system equations of motion are given by [4]

$$\begin{cases} r\ddot{\varphi} + 2\dot{r}\,\dot{\varphi} = -a_M + a_T \sin\varphi \\ \ddot{r} - r\dot{\varphi}^2 = -a_T \cos\varphi \end{cases} \tag{2}$$

where: φ is the bearing angle, which is measured counter-clockwise from the target velocity direction; r is the range between missile and target; a_M is the missile acceleration (control input), assumed to be perpendicular to the line-of-sight (LOS); and a_T is the target acceleration. The white noise corrupted passive (bearing angle only) seeker data is available

$$y(t) = \varphi + w(t) \tag{3}$$

where $w(t)$ is white noise.

The appropriate guidance law for the engagement model given by Eq.(2) is derived with the help of the Lyapunov stability theory [4]

$$a_M = a_T \sin\varphi - \lambda \dot{r}\dot{\varphi} \tag{4}$$

where λ is the navigation gain, which assumed to be a constant.

This guidance law belongs to the family of Augmented True Proportional Navigation guidance laws. Their basic principles are to take into account the target acceleration term, and to apply the commanded lateral acceleration a_M perpendicular to LOS. In order to exploit this guidance law, an extended Kalman filter is employed as the state observer of the linearized system to obtain the estimations of the bearing angle and range rates. For this

purpose, the nonlinear model of the engagement process Eq.(2) is linearized and presented in Cartesian coordinates by the following system equations:

$$\begin{cases} \dot{z}_1 = z_3 \\ \dot{z}_2 = z_4 \\ \dot{z}_3 = a_T - a_{M_1} \\ \dot{z}_4 = -a_{M_2} \end{cases} \tag{5}$$

The associated nonlinear measurement equation is

$$y(t) = \arctan(\frac{z_2}{z_1}) + w(t) \tag{6}$$

The algebraic relations, similar to the identical pair of Eq.(3) and Eq.(6), connect the EKF state Eq.(5) estimations with the terms of the guidance law Eq.(4).

4 Formal Verification of the Realistic Interception Scenario

Formal methods verify only discrete systems and protocols. Thus, continuous-time system Eq.(2) was transformed into a periodically updated system, similar to the Matlab presentation of the continuous-time systems. In our case, we described the overall closed-loop system Eqs.(2-6) in the Verilog language [9], which has a powerful compiler to synthesize the Verilog model into a logical circuit of basic logical gates. All numbers were represented by 32-bit vectors.

The tracking algorithm was applied to a realistic interception scenario [4], presented in Table 1. The system properties were verified with the help of RuleBase, a formal verification tool developed by the IBM Haifa Research Laboratory.

Table 1. The initial conditions of the scenario and system's parameters.

LOS angle rate	$\dot{\varphi}$	$\frac{rad}{sec}$	0.0051
LOS angle	φ	rad	-0.9600
range rate	$\dot{r}$	$\frac{m}{sec}$	-1787
range	r	m	80000
target velocity	V_T	$\frac{m}{sec}$	500
missile velocity	V_M	$\frac{m}{sec}$	1500
navigation constant	λ		3.85
measurement noise variance	1σ	rad^2	$8 \cdot 10^{-8}$
process sampling rate		sec	0.5
EKF update rate		sec	1.0

The first basic property we checked was the following one: Consider the target acceleration to be a white noise signal bounded in amplitude by 30 $\frac{m}{sec^2}$. Suppose that after $k = 5$ seconds

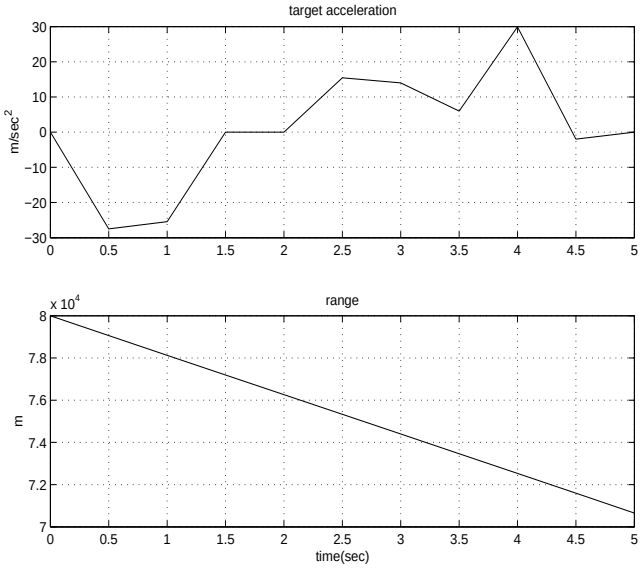

Figure 2: Property 1: target acceleration and range profiles.

from the start of interception process, the distance r between the missile and target must always decrease from the initial 80000 meters to less than 70000 meters. The specially-chosen gains of the controller (navigation constant λ) must ensure this. The simulation procedure usually used to check the controller consistency is to launch at least 500 Monte Carlo trials. RuleBase found a counterexample in a single run approximately 300 seconds after verifying the following property

Property 1:

$$\{\mathbf{always}(range = 80000 \rightarrow \mathbf{next}[k](range < 70000))\} \tag{7}$$

relative to all possible perturbations of target acceleration. The counterexample provides the target acceleration profile (Figure 2) that caused the range to be at least 71000 meters after the first 5 seconds of the interception process.

The second property we formulated was the following: The missile acceleration a_M cannot exceed $20\frac{m}{sec^2}$. This property can be formulated in two ways

Property 2.1:

$$\{\mathbf{always}\ (a_M < 20)\} \tag{8}$$

Property 2.2:

$$\{\mathbf{exists\_at\_least\_once}\ (a_M > 20)\} \tag{9}$$

Here, RuleBase found a counterexample showing a target acceleration that causes the missile acceleration to be $24\frac{m}{sec^2}$ after the first 4 seconds of the interception process.

The next group of properties we checked dealt with a bias in the EKF range and range rate estimations. It is well known that given only noisy measurements of the bearing (line-of-sight) angle, EKF frequently provides biased range and range rate output estimations [3]. Formal methods can help analyze these effects. The provided counterexamples detect

all particular target maneuvers that caused and amplified the biases. For our example, we considered the initial range bias to be 4000 meters (i.e., the 5% of the initial range). Next, we found the target acceleration that provides such a range bias in the first ten seconds of the interception process. For this purpose, the Rulebase was launched in BMC mode for $k = 10$ seconds of the interception process. The corresponding property has the following form

Property 3:

$$\{\mathbf{always}\,[k]\,(range\_bias < 4000)\} \tag{10}$$

The results are shown in Figure 3: the target acceleration, range and range rate are depicted by a solid line. Missile acceleration and biased range and range rate estimations are depicted by a dashed line.

The EKF can be unbiased by improving the correlation between the EKF gain, and innovation sequences against this particular target maneuver and all possible target maneuvers in principal [2],[3]. The most straightforward way to do this is to add an artificial process noise to compensate for the errors in the state predicition [3], by using a larger modified process noise covariance in the covariance prediction equation. The modified covariance is presented by

$$Q_m(k) = Q_p(k) + Q(k) \geq Q(k) \tag{11}$$

where $Q_p(k)$ is the positive semidefinite pseudo-noise covariance, and $Q(k)$ is a process noise covariance of the additive, zero mean and white process noise $v(k)$

$$E[v(k)] = 0 \tag{12}$$

$$E[v(k)v(j)^T] = Q(k)\delta_{kj} \tag{13}$$

Such increase of the state covariance will cause a state gain to be larger, thus giving more weight to the recent data. This technique has a completely heuristic nature, and is reccomended for use under the assumption that linearization errors are zero-mean and white [3]. If this assumption does not hold, this method will probably not accomplish much. However any small improvement is still a definite achievement in such a complicated case.

The formal verification easily finds the required heuristic pseudo-noise covariance matrix (gain) $Q_p(k)$ by checking the following property: Consider $Q_p(k) \in [0..0.01]$, and check that for all $Q_p(k)$ from this interval the range bias is always larger than 4000 meters on the time interval of k cycles. Restrict the target maneuver to be the same as that obtained from verification of the previous property.

Property 4.1:

$$\{\mathbf{always}\ (\mathbf{for\_all}(Q_p(k) \in [0..0.01]) \rightarrow \mathbf{within}[0..k]\ range\_bias > 4000)\} \tag{14}$$

A stronger variant of Property 4.1 includes a range rate bias term

Property 4.2:

$$\{\mathbf{always}\ (\mathbf{for\_all}(Q_p(k) \in [0..0.01]) \rightarrow \mathbf{within}[0..k]\ range\_bias > 4000 \wedge range\_rate\_bias > 400)\} \tag{15}$$

The obtained counterexample provides the pseudo-noise covariance matrix (gain) $Q_p(k)$ that decrease the bias by 2500 meters in the first $k = 10$ seconds of the run. The results are depicted in Figure 3 by the dashedot line.

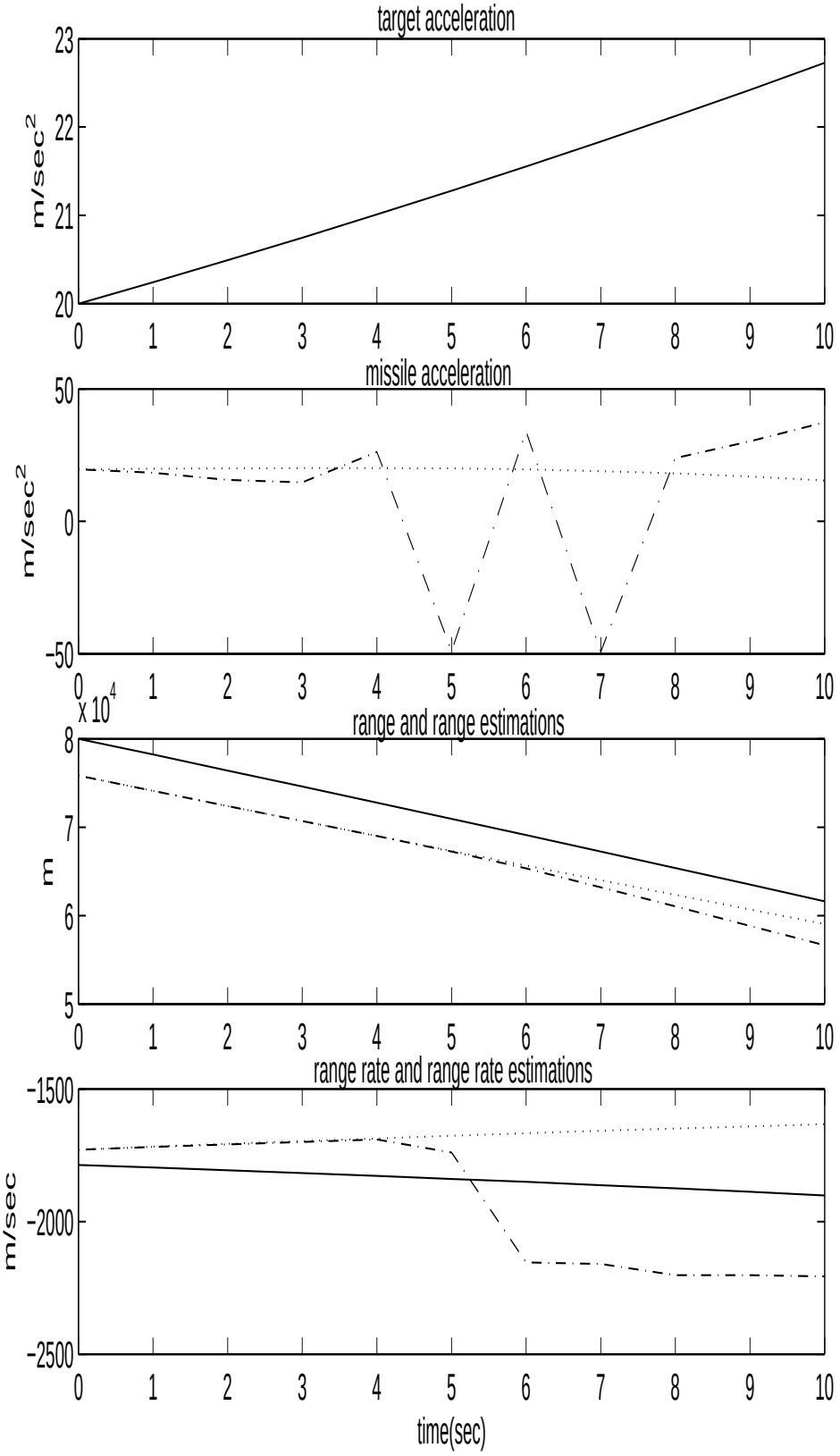

Figure 3: Properties 3 and 4: interception parameters.

5 Conclusions and Future Research

The main contribution of this work is the application of formal verification methods to the target estimation algorithms under a realistic air-to-air interception planar scenario. A novel powerful technique is introduced to analyze the interception process behavior, and in particular the effect of EKF range and range rate estimation bias. The system properties are naturally described in a formal specification language. The RuleBase verification engine steadily verifies these properties in BMC mode. An average property needs 300-900 seconds to run on a Linux machine. The system implementation in Verilog customizes a floating-point arithmetic processing.

The verification protocol of the illustrated example shows that formal verification techniques are capable of finding heuristic control parameters, and proved to be suitable for checking the bound and corner cases. In general, the proposed approach will be extremely efficient for checking the consistency of the switching control algorithms, fuzzy-logic-based control, and design of controllers for actuators with quasi-stable dynamics. These issues will be the subjects of our future research.

References

[1] E.M.Clarke, O. Grumberg, and D.A. Peled. *Model Checking.* The MIT Press, 1999.

[2] Maybeck P.S.*Stochastic Models, Estimation and Control.* Academic Press Inc.,1982.

[3] Y. Bar-Shalom, T. Kirubarajan, Xiao-ro Li. *Estimation with Applications to Tracking and Navigation.* Wiley-Interscience, 2001.

[4] M. Moulin, E. Kreindler, and M. Guelman. "Ballistic Missile Interception with Bearings-Only Measurements." *Proc. of the Israel Conf. on Aeronautics and Astronautics*, 1996.

[5] E.M. Clarke, A. Bierre, R. Raimi,and Zhu Y. "Bounded Model Checking Using Satisfiability Solving". *Tools and Algorithms for the Analysis and Construction of Systems (TACAS'99)* , number 1579 in LNCS, Springer-Verlag, 1999.

[6] I. Beer, S. Ben-David, C. Eisner, D. Geist, L. Gluhovsky, T. Heyman, A. Landver, P. Paanah, Y. Rodeh, G. Ronin, and Y. Wolfsthal. "RuleBase: Model Checking at IBM," in Proc. 9 th International Conference on Computer Aided Verification (CAV), LNCS 1254. Springer-Verlag, 1997.

[7] I. Beer, S. Ben-David, C. Eisner, and A. Landver., "RuleBase: An Industry-oriented Formal Verification Tool," In Proc. 33 rd Design Automation Conference (DAC), pages 655-660. Association for Computing Machinery, Inc., June 1996.

[8] RuleBase-Formal Verification (FV) Tool, developed by the IBM Haifa Research Laboratory: "http://www.haifa.il.ibm.com/projects/verification/RB_Homepage".

[9] S. Palnitkar. *Verilog HDL.* Prentice Hall PTR, 1996.