**DEVS-based modelling**

**for radar-augmented MANPADS engagement**

by

Françoise Blanc

Student ID: 101209287

Submitted to: Professor Gabriel A. Wainer

Assignment 1  
SYSC 5104 – Methodologies for Discrete Event Modelling and Simulation

Department of Systems and Computer Engineering

Faculty of Engineering

Carleton University

October 23, 2020

**CONTENT**

[PART 1 – CONCEPTUAL MODEL 3](#_Toc54336822)

[1. CONTEXT 3](#_Toc54336823)

[2. PROBLEM STATEMENT 3](#_Toc54336824)

[3. MODEL OVERVIEW 3](#_Toc54336825)

[4. MODEL ASSUMPTIONS 4](#_Toc54336826)

[PART 2 – DEVS FORMAL SPECIFICATIONS 5](#_Toc54336827)

[1. ATOMIC MODELS 5](#_Toc54336828)

[Aircraft = <X, Y, S, δext, δint, λ, ta> 5](#_Toc54336829)

[Radar = <X, Y, S, δext, δint, λ, ta> 7](#_Toc54336830)

[Gunner = <X, Y, S, δext, δint, λ, ta> 8](#_Toc54336831)

[Missile = <X, Y, S, δext, δint, λ, ta> 10](#_Toc54336832)

[2. COUPLED MODELS 11](#_Toc54336833)

[Air Defence System (ADS) = < X, Y, M, EIC, EOC, IC, SELECT > 11](#_Toc54336834)

[Radar-Augmented MANPADS Engagement = < X, Y, M, EIC, EOC, IC, SELECT > 11](#_Toc54336835)

[PART 3 – MODEL EXPERIMENTS 12](#_Toc54336836)

[1. ATOMIC MODEL EXPERIMENTS 13](#_Toc54336837)

[Aircraft Testing 13](#_Toc54336838)

[Radar Testing 15](#_Toc54336839)

[Gunner Testing 16](#_Toc54336840)

[Missile Testing 25](#_Toc54336841)

[2. COUPLED MODEL EXPERIMENTATION 26](#_Toc54336842)

[Air Defence System Testing 26](#_Toc54336843)

[3. INTEGRATION TESTING 28](#_Toc54336844)

[Integration Test #1 28](#_Toc54336845)

[Integration Test #2 32](#_Toc54336846)

[PART 4 – DEVS WEB VIEWER 34](#_Toc54336847)

[PART 5 – FUTURE WORK 34](#_Toc54336848)

# PART 1 – CONCEPTUAL MODEL

## CONTEXT

Military aircraft deployed in theatre can encounter various threats. As new threats and new technologies are being developed, countries need to address these challenges to ensure military aircraft protection readiness. One way to assess aircraft protection is through the use of modelling and simulation.

## PROBLEM STATEMENT

One major threat against military aircraft in conflicts involving non-state armed groups (NSAG) is man-portable air defence system (MANPADS). MANPADS, such as the one shown in Figure 1, are short-range surface-to-air missile systems intended for low-flying aircraft. These systems are usually designed to be shoulder-launched by one single person. MANPADS are often the weapons of choice to NSAG as they are widely proliferated, relatively cheap, and easy to transport and use.



Figure : MANPADS missile launch[[1]](#footnote-1)

One limitation of MANPADS is the requirement to acquire the target visually. Cueing devices, such as radar providing target position well beyond the visual acquisition range, are now being incorporated into MANPADS and used by NSAG. The use of DEVS modelling is being investigated to assess the effectiveness of such engagement using radar-augmented MANPADS.

## MODEL OVERVIEW

The DEVS conceptual model representing a radar-augmented MANPADS engagement is shown in Figure 2. This model consists of two main components: an *Aircraft* and an *Air Defence System*. The *Air Defence System* component is further divided into the following components: *Radar*, *MANPADS Gunner* and *Missile*.

The simulation starts once an aircraft enters a mission area with a predefined path.

* **Aircraft:** The aircraft model generates the *aircraft position* of a predefined path every second while a missile does not hit the aircraft.
* **Radar:** The radar model has two main phases: *searching* or *detected*. Once the radar detects the aircraft within a predefined range (default value set to 20 km), it relays the *detected position* every second to the MANPADS gunner.
* **MANPADS Gunner:** The gunner model has four main phases: *waiting*, *launching, loading and hit*. Once the gunner receives a *detected position* from the radar, the gunner switches to *launching* if the aircraft is within a predefined range from the gunner (default value is set to 6 km) and then starts generating a *distance* (i.e., distance between the gunner and the aircraft). The gunner also receives information about the *missile status* (i.e., hit or miss). If a missile misses the aircraft, the gunner starts *loading* a new missile for 40 seconds and then starts *waiting* for the next *detected position*. The gunner has 4 missiles (default value).
* **Missile:** The missile model has two main phases: *waiting* or *launched*. The missile receives a *distance* from the MANPADS gunner and then determines the *missile status* (i.e., hit or miss). The probability of hit depends on the distance between the missile and the aircraft at missile launch. The *missile status* is relayed to the aircraft and gunner model.

The simulation ends once the aircraft has flown its predefined path or has been hit by a missile.

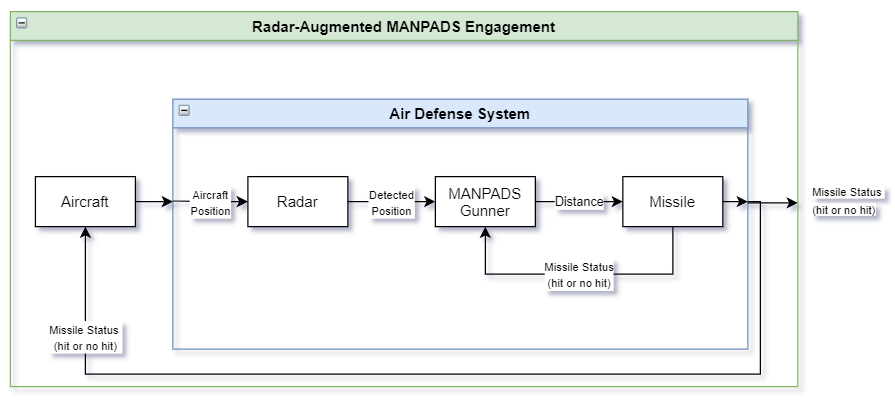


Figure : Conceptual model

## MODEL ASSUMPTIONS

To limit the extent of the model, the following assumptions have been made:

* The mission area consists of one aircraft, one radar and one MANPADS gunner;
* The mission area is a flat terrain (no obstruction) and sky clear.
* The aircraft maintains a predefined straight path at constant altitude and speed without evasive manoeuvre;
* The radar detection range is not target-dependent and set to 20 km;
* The maximum missile range is set to 6 km; and
* The probability of hit only depends on the distance between the missile and the aircraft.

# PART 2 – DEVS FORMAL SPECIFICATIONS

## ATOMIC MODELS

### Aircraft = <X, Y, S, δext, δint, λ, ta>

|  |  |
| --- | --- |
| **State Variables** | **Formal Specification** |
| Sigma = 0  Phase = Flying  Aircraft Initial Position = [-30000,0] (in meters)  Aircraft Final Position = [30000,0] (in meters)  Aircraft Speed = 60 m/s  Aircraft Position  Flight Path Distance  Distance Flown  Direction Vector  Missile Status | X = {Missile Status}  Y = {Aircraft Position}  S = {Sigma, Phase, Aircraft Initial Position, Aircraft Final Position, Aircraft Speed, Aircraft Position, Flight Path Distance, Distance Flown, Direction Vector, Missile Status} |
| **Functions** | |
| **δext (s, e, x = Missile Status)** {  if (Distance Flown < Flight Path Distance) {  if (Missile Status == 1 (hit)) {  Phase = Hit;  Sigma = ∞;  } else { // Missile Status == 0 (miss)  Phase = Flying;  Sigma = 1 second  }  }  }  **δint (s)** {  Calculate new Aircraft Position;  Calculate Distance Flown;  if (Distance Flown > Flight Path Distance) {  Phase = Mission completed;  Sigma = ∞;  } else {  Phase = Waiting;  Sigma = 1 second;  }  }  **λ(s)** {  Send Aircraft Position to the port *out;*  } | |
| **Experimentation Strategy** | |
| To ensure the correctness of the aircraft atomic model, the test cases verifies that the model:   1. generates an aircraft position every second; 2. stops generating an aircraft position once the aircraft has completed its predefined flight path; 3. stops generating an aircraft position once a Missile Status of 1 (hit) is received; and 4. continues generating aircraft position once a Missile Status of 0 (miss) is received. | |

### Radar = <X, Y, S, δext, δint, λ, ta>

|  |  |
| --- | --- |
| **State Variables** | **Formal Specification** |
| Sigma = ∞  Phase = Searching  Aircraft Position  Detected Position  Radar Position = [0,0] (in meters)  Radar Range = 20000 m  Distance Radar Aircraft | X = {Aircraft Position}  Y = {Detected Position}  S = {Sigma, Phase, Aircraft Position, Detected Position, Radar Position, Radar Range, Distance Radar Aircraft} |
| **Functions** | |
| **δext (s, e, x = Aircraft Position)** {  Calculate Distance Radar Aircraft;  if (Distance Radar Aircraft ≤ 20 km) {  Phase = Detected;  Detected Position = Aircraft Position;  Sigma = 1 seconds;  } else {  Phase = Searching;  Sigma = ∞;  }  }  **δint (s)** {  case Phase  Detected:  Phase = Searching;  Sigma = ∞;  }  **λ(s)** {  if (Phase == Detected) {  Send Detected Position to the port *out;*  }  } | |
| **Experimentation Strategy** | |
| To ensure the correctness of the radar atomic model, the test cases verify that the model:   1. generates an aircraft position (*detected position*) once an aircraft position is received and is within 20 km from the radar. | |

### Gunner = <X, Y, S, δext, δint, λ, ta>

|  |  |
| --- | --- |
| **State Variables** | **Formal Specification** |
| Sigma = ∞  Phase = Waiting  Detected Position  Gunner Range  Missile Reload Time = 40 seconds  Reload Time  Missile Status  Distance Gunner Aircraft  Number Missiles = 4  Gunner Position = [-10000,0] in meters | X = {Detected Position, Missile Status}  Y = {Distance Gunner Aircraft}  S = {{Sigma, Phase, Detected Position, Gunner Range, Missile Reload Time, Reload Time Missile Status, Distance Gunner Aircraft, Number Missiles, Gunner Position}} |
| **Functions** | |
| **δext (s, e, x = {Detected Position, Missile Status})** {  if x = Missile Status {  if (Phase == Loading) {  Sigma = Reload Time;  Reload Time = Sigma;  }  if ((Phase == Waiting) && (Distance Gunner Aircraft ≤ 6 km)) {  if (Missile Status == 1) {  Phase = Hit;  Sigma = ∞;  } else if (Missile Status == 0) {  Number Missiles -= 1;  if (Number Missiles > 0) {  Phase = Loading;  Sigma = Missile Reload Time;  Reload Time = Sigma;  } else {  Phase = Waiting;  Sigma = ∞;  }  }  }  }  if x = Detected Position {  if (Phase == Loading) {  Sigma = Reload Time – e;  Reload Time = Sigma;  }  if (Phase == Waiting) {  Calculate Distance Gunner Aircraft;  if (Distance Gunner Aircraft ≤ 6 km) && (Number Missiles > 0) {  Phase = Launching;  Sigma = 0;  } else {  Phase = Waiting;  Sigma = ∞;  }  }  }  }  **δint (s)** {  case Phase  Launching:  Phase = Waiting;  Sigma = ∞;  Loading:  Phase = Waiting;  Sigma = ∞;  }  **λ(s)** {  if Phase = Launching {  Send Distance Gunner Aircraft to the port *out*;  }  } | |
| **Experimentation Strategy** | |
| To ensure the correctness of the gunner atomic model, the test cases verify that the model:   1. only generates a distance (i.e., the distance between the gunner and the aircraft) once the aircraft is within 6 km; 2. does not generate any output when the model receives a missile status outside of 6 km; 3. does not relaunch another missile for 40 seconds (i.e., missile reload time) once a missile has missed the aircraft; 4. disregards missile status received during loading; and 5. does not generate outputs once the gunner has launched a total of 4 missiles. | |

### Missile = <X, Y, S, δext, δint, λ, ta>

|  |  |
| --- | --- |
| **State Variables** | **Formal Specification** |
| Sigma = ∞  Phase = Waiting  Distance Missile Aircraft  Missile Status  Missile Flying Time | X = {Distance Gunner Aircraft}  Y = {Missile Status}  S = {{Sigma, Phase, Distance Missile Aircraft, Missile Status}} |
| **Functions** | |
| **δext (s, e, x = Distance Gunner Aircraft)** {  if (Phase == Waiting) {  if (Distance Missile Aircraft ≤ 6 km) {  Phase = Launched;  if (Distance Gunner Aircraft ≤ 6 km) && (Distance Gunner Aircraft ≥ 4 km) {  Missile Status = (rand() % 100) < 30;  // Set to 30 for the tests. This value can be changed by the user.  Missile Flying Time = 7 seconds;  Sigma = Missile Flying Time;  } else if (Distance Gunner Aircraft < 4 km) && (Distance Gunner Aircraft ≥ 1 km) {  Missile Status = (rand() % 100) < 30;  // Set to 30 for the tests. This value can be changed by the user.  Missile Flying Time = 5 seconds;  Sigma = Missile Flying Time;  } else {  Missile Status = (rand() % 100) < 30;  // Set to 30 for the tests. This value can be changed by the user.  Missile Flying Time = 5 seconds;  Sigma = Missile Flying Time;  }  } else {  Phase = Waiting;  Sigma = ∞;  }  **δint (s)** {  case Phase  Launched:  Phase = Waiting;  Sigma = ∞;  }  **λ(s)** {  if Phase = Launched {  Send Missile Status to the port *out*;  }  } | |
| **Experimentation Strategy** | |
| To ensure the correctness of the missile atomic model, the test cases verify that the model:   1. generates a missile status (i.e., hit or miss) only when the model receives a distance between the gunner and the aircraft; and 2. generates a missile status after the missile flying time delay. | |

## COUPLED MODELS

### Air Defence System (ADS) = < X, Y, M, EIC, EOC, IC, SELECT >

|  |  |
| --- | --- |
| X: | {Aircraft Position} |
| Y: | {Missile Status} |
| M: | {Radar,  Gunner,  Missile} |
| EIC: | {ADS.inAircraftPosition, Radar.inAircraftPosition} |
| EOC: | {Missile.outMissileStatus, ADS.outMissileStatus} |
| IC: | {(Radar.outDetectedPosition,Gunner.inDetectedPosition),  (Gunner.outDistanceGunnerAircraft,Missile.inDistanceGunnerAircraft),  (Missile.outMissileStatus, Gunner.inMissileStatus) |
| SELECT: | {Radar, Gunner, Missile} = Missile  {Radar, Gunner} = Gunner |
| **Experimentation Strategy** | |
| To ensure the correctness of the ADS coupled model, the test cases verify that the model:   1. generates proper missile statuses. | |

### Radar-Augmented MANPADS Engagement = < X, Y, M, EIC, EOC, IC, SELECT >

|  |  |
| --- | --- |
| X: | {} |
| Y: | {Missile Status} |
| M: | {Aircraft,  Air Defence System} |
| EIC: | {}, |
| EOC: | {ADS.outMissileStatus, radarMANPADSengagement.outMissileStatus} |
| IC: | {(Aircraft.outAircraftPosition, ADS.inAircraftPosition),  (ADS.outMissileStatus, Aircraft.inMissileStatus)} |
| SELECT: | {Aircraft, Air Defence System} = Air Defence System |
| **Experimentation Strategy** | |
| Run different mission scenarios (e.g., changing the flight path, number of missiles, etc.) to ensure that the model adheres to the expected behaviours. | |

# PART 3 – MODEL EXPERIMENTS

Each atomic and coupled model has been tested individually using a simple mission scenario depicted in Figure 3. The aircraft flies a predefined straight flight path from initial position [‑30 000, 0] m to final position [30 000, 0] at a constant speed of 60 m/s. The aircraft does not conduct any evasive manoeuvre once targeted by a missile. A radar is positioned at [0, 0] and it relays position details to a gunner located at 10 km from the radar. The gunner position is at [‑10 000, 0]. Radar detection range and missile kinematic range are set to 20 and 6 km, respectively. The gunner has 4 missiles available. The probability of a successful hit at any missile launch ranges has been set to 30%. The scenario parameters are summarized in Table 1.

Individual tests are performed at the atomic level, first starting with the aircraft atomic model up to the missile atomic model and then, the coupled tests are performed.

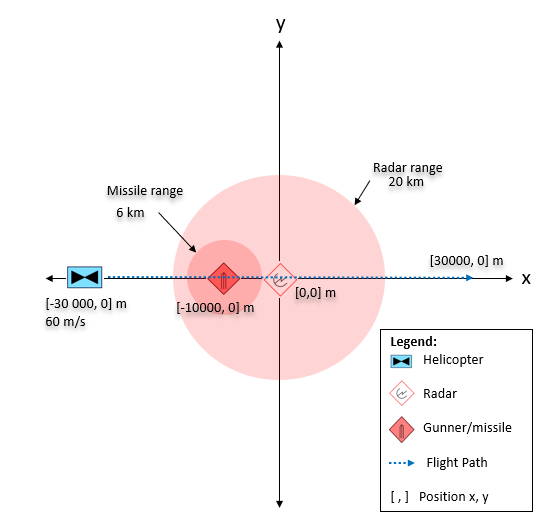


Figure : Mission scenario setup for individual tests

Table : Scenario parameters

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Description** | **Default Value** |
| **Aircraft atomic model** | | |
| selectedAircraftInitialPositionX | Aircraft initial position in x in meters | -30 000 m |
| selectedAircraftInitialPositionY | Aircraft initial position in y in meters | 0 |
| selectedAircraftFinalPositionX | Aircraft final position in x in meters | 30 000 m |
| selectedAircraftFinalPositionY | Aircraft final position in y in meters | 0 |
| selectedAircraftSpeed | Aircraft speed in meters per second | 60 m/s |
| **Radar atomic model** | | |
| radarPositionX | Radar position x in meters | 0 m |
| radarPositionY | Radar position y in meters | 0 m |
| radarRange | Radar detection range in meters | 20 000 m |
| **Gunner atomic model** | | |
| gunnerPositionX | Gunner position x in meters | -10 000 m |
| gunnerPositionY | Gunner position x in meters | 0 |
| missileNumber | Number of missiles available for the gunner | 4 |
| **Missile atomic model** | | |
| poh\_4\_6 | Probability of hit for missile launch range between 4 and 6 km | 30 |
| poh\_1\_4 | Probability of hit for missile launch range between 1 and 4 km | 30 |
| poh\_0\_1 | Probability of hit for missile launch range less than 1 km | 30 |

## ATOMIC MODEL EXPERIMENTS

### Aircraft Testing

|  |
| --- |
| **Test Case #1** |
| **Executable:** AIRCRAFT\_TEST.exe |
| **Description** |
| * The aircraft atomic model must generate an aircraft position every second. * The aircraft atomic model stops generating aircraft position once the aircraft has completed its predefined flight path. |
| **Test Inputs** |
| * The parameters of the atomic model are set to the values shown in Table 1. * Simulation time is set to 20 minutes. * No input data file is required for this test case as this model auto generates an aircraft position every second. * **Input data file:** \input\_data\aircraft\_input\_test.txt (empty file) |
| **Test Ouputs** |
| **Output data files:**   * \simulation\_results\aircraft\_test\_output\_messages\_test\_case1.txt * \simulation\_results\aircraft\_test\_output\_state\_test\_case1.txt   Beginning of simulation results log:    End of simulation results log: |
| **Discussion** |
| The simulation results demonstrate that an aircraft position is outputted every second and that the model no longer generates a position once the aircraft has reached the end of it predefined path (i.e., [30 000, 0]). The simulation stops at time 00:16:39. |

|  |
| --- |
| **Test Case #2** |
| **Executable:** AIRCRAFT\_TEST.exe |
| **Description** |
| * The aircraft atomic model must stop generating aircraft position once a Missile Status of 1 (hit) is received. * The aircraft atomic model must continue generating aircraft position once a Missile Status of 0 (miss) is received. |
| **Test Inputs** |
| * The parameters of the atomic model are set to the values shown in Table 1. * Simulation time is set to 20 minutes. * **Input data file:** \input\_data\aircraft\_input\_test\_t2.txt |
| **Test Ouputs** |
| **Output data files:**   * \simulation\_results\aircraft\_test\_output\_messages\_test\_case2.txt * \simulation\_results\aircraft\_test\_output\_state\_test\_case2.txt   […]    […] |
| **Discussion** |
| * The simulation results show that once the model has received a missile status of 0 (miss) at time 00:06:00, the atomic model continues to generate aircraft position. The results also show that once the aircraft has been hit, meaning that it has received a missile status of 1 as shown at time 00:07.00, the atomic model stops generating an aircraft position. The simulation ends. |

### Radar Testing

|  |
| --- |
| **Test Case #1** |
| **Executable:** RADAR\_TEST.exe |
| **Description** |
| * The radar atomic model must only generate an aircraft position (*detected position*) once an aircraft position is received and is within 20 km from the radar. |
| **Test Inputs** |
| * The parameters of the atomic model are set to the values shown in Table 1. * Simulation time is set to 20 minutes. * **Input data file:** \input\_data\radar\_input\_test.txt   Aircraft positions:    […] |
| **Test Ouputs** |
| **Output data files:**   * \simulation\_results\radar\_test\_output\_messages\_test\_case1.txt * \simulation\_results\radar\_test\_output\_state\_test\_case1.txt   […]    […] |
| **Discussion** |
| * The simulation results show that the radar model only generates an aircraft position once the aircraft is within the detection range of the radar, which is set to 20 km. In this test case, the radar model outputs a detected position every second from 00:02:48 to 00:13:54. |

### Gunner Testing

|  |
| --- |
| **Test Case #1** |
| **Executable:** GUNNER\_TEST.exe |
| **Description** |
| * Verify that the output (distance between the gunner and the aircraft) is generated once the aircraft is within 6 km. |
| **Test Inputs** |
| * The parameters of the atomic model are set to the values shown in Table 1. * Simulation time is set to 20 minutes. * **Input data files:**    + \input\_data\gunner\_detected\_position\_input\_test.txt   + \input\_data\gunner\_missile\_status\_input\_test.txt (empty file)   This gunner atomic model has two inputs: the detected aircraft position from the radar and the missile atomic model’s missile status. As a reminder of the mission scenario, the aircraft moves along the x-axis towards the radar positioned at [0,0]. The gunner is located at [‑10000, 0], which is between the aircraft initial position [-30 000,0] and the radar position [0,0]. In this test case, no missile status inputs are provided; only detected aircraft positions are provided.  Detected aircraft positions:    […] |
| **Test Ouputs** |
| **Output data files:**   * \simulation\_results\gunner\_test\_output\_messages\_test\_case1.txt * \simulation\_results\gunner\_test\_output\_state\_test\_case1.txt   […]    […]    […] |
| **Discussion** |
| * Considering the provided inputs, the gunner atomic model should only generate an output (i.e., outDistanceGunnerAircraft - distance between the gunner and the aircraft) when the aircraft is within [-16000, 0] and [-4000,0] along the x-axis. * The results provided above show that the gunner atomic model has the expected behaviour: the model only generates an output once the aircraft is within the missile kinematic range set to 6 km. In this test case, the gunner atomic model generates an output every second from time 00:03:55 to 00:07:14. |

|  |
| --- |
| **Test Case #2** |
| **Executable:** GUNNER\_TEST.exe |
| **Description** |
| * Verify that no output is generated when the gunner receives a missile status outside of 6 km. |
| **Test Inputs** |
| * The parameters of the atomic model are set to the values shown in Table 1. * Simulation time is set to 20 minutes. * **Input data files:**    + \input\_data\gunner\_detected\_position\_input\_test.txt   + \input\_data\gunner\_missile\_status\_input\_test2.txt   In this test case, a missile status (either 0 or 1) is provided while the aircraft is not within 6 km of the gunner.  Detected aircraft positions:    […]    Missile status: |
| **Test Ouputs** |
| **Output data files:**   * \simulation\_results\gunner\_test\_output\_messages\_test\_case2.txt * \simulation\_results\gunner\_test\_output\_state\_test\_case2.txt   […]    […] |
| **Discussion** |
| * The simulation results show that no outputs are generated for missile status received outside of the gunner range. |
| **Test Case #3** |
| **Executable:** GUNNER\_TEST.exe |
| **Description** |
| 1. Verify that once a missile has missed the aircraft, no other missile is launch within 40 seconds (i.e., missile reload time). 2. Verify that if a missile status is received during loading, the input is disregarded and does not have an impact on the simulation results. 3. Verify that once the gunner has launched a total of 4 missiles, the gunner no longer generates outputs. |
| **Test Inputs** |
| * The parameters of the atomic model are set to the values shown in Table 1. * Simulation time is set to 20 minutes. * **Input data files:**    + \input\_data\gunner\_detected\_position\_input\_test.txt   + \input\_data\gunner\_missile\_status\_input\_test3.txt   Detected aircraft positions:    […]  Missile status: |
| **Test Ouputs** |
| **Output data files:**   * \simulation\_results\gunner\_test\_output\_messages\_test\_case3.txt * \simulation\_results\gunner\_test\_output\_state\_test\_case3.txt   […]    **Output state at time 00:04:00:**    […]    […]    **Output state at time 00:04:50:**    […]    **Output state at time 00:05:40:**    […]    **Output state at time 00:06:30:**    […] |
| **Discussion** |
| * At time 00:04:00, the gunner receives a missile status of 0 (miss) and stops generating outputs for the next 40 seconds corresponding to the missile reload time. The gunner starts generating again outputs at 00:04:39, which is a second earlier than the expected behaviour. This discrepancy remains to be fixed. Also, note that at time 00:04:05, the missile status of 0 received during loading was disregarded and did not affect the simulation. * At time 00:04:50, 00:05:40 and 00:06:30, the gunner model receives a missile status of 0. The output states at time 00:04:50 and 00:05:40 indicate that the gunner is now “LOADING” and still has missiles. However, at time 00:06:30, the gunner is now “WAITING” and not generating any outputs since the gunner no longer has missile. These results represent the expected behaviour of the gunner model. |

|  |
| --- |
| **Test Case #4** |
| **Executable:** GUNNER\_TEST.exe |
| **Description** |
| 1. Verify that the gunner atomic model stops generating outputs once it receives a missile status of 1 (hit). |
| **Test Inputs** |
| * The parameters of the atomic model are set to the values shown in Table 1. * Simulation time is set to 20 minutes. * **Input data files:**    + \input\_data\gunner\_detected\_position\_input\_test.txt   + \input\_data\gunner\_missile\_status\_input\_test4.txt   Detected aircraft positions:    […]  Missile status: |
| **Test Ouputs** |
| **Output data files:**   * \simulation\_results\gunner\_test\_output\_messages\_test\_case4.txt * \simulation\_results\gunner\_test\_output\_state\_test\_case4.txt   […]    […] |
| **Discussion** |
| * The simulations results show that once the gunner model receives a missile status of 1 (hit), the model stops generating outputs. |

### Missile Testing

|  |
| --- |
| **Test Case #1** |
| **Executable:** MISSILE\_TEST.exe |
| **Description** |
| * The missile atomic model must generate a missile status (i.e., hit or miss) when it receives a distance between the gunner and the aircraft. The missile status is outputted after a delay representing the missile flying time which is based on the missile launch range. |
| **Test Inputs** |
| * The parameters of the atomic model are set to the values shown in Table 1. * Simulation time is set to 20 minutes. * **Input data file:** \input\_data\missile\_input\_test.txt   Distance |
| **Test Ouputs** |
| **Output data files:**   * \simulation\_results\missile\_test\_output\_messages\_test\_case1.txt * \simulation\_results\missile\_test\_output\_state\_test\_case1.txt   Output Message:    Output State: |
| **Discussion** |
| * The simulation results meet the expected behaviour. |

## COUPLED MODEL EXPERIMENTATION

### Air Defence System Testing

|  |
| --- |
| **Test Case #1** |
| **Executable:** ADS\_TEST.exe |
| **Description** |
| * Ensure that the Air Defence System coupled model, which consists of the radar, gunner and missile atomic models, generates missile statuses. |
| **Test Inputs** |
| * The parameters of the atomic model are set to the values shown in Table 1. * Simulation time is set to 20 minutes. * **Input data file:** \input\_data\ads\_input\_test.txt   Aircraft positions:    […] |
| **Test Ouputs** |
| **Output data files:**   * \simulation\_results\ads\_test\_output\_messages\_test\_case1.txt * \simulation\_results\ads\_test\_output\_state\_test\_case1.txt   […]    […] |
| **Discussion** |
| * As shown in the simulation results at time 00:04:46, the coupled model correctly generated a missile status of 1 (hit). |

## INTEGRATION TESTING

The Radar-Augmented MANPADS Engagement model allows the user to test various mission scenarios. Table 1 presents the scenario parameters that can be modified. The last column of the table presents the default values used for the atomic and coupled model tests.

### Integration Test #1

As a first experiment, a different flight path and a different gunner position have been selected. This new mission scenario is depicted in Figure 4. Additionally, to test the gunner’s reload function once a missile misses the aircraft, let’s assume that the gunner has limited experience in the use of MANPADS. The probability of hit for all ranges has been reduced to 20%.

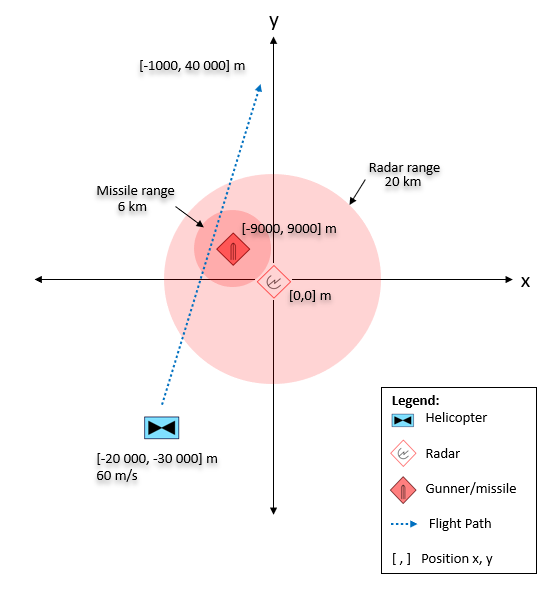


Figure : Mission scenario setup for integration test #1

**Input data file:**

* \input\_data\engagement\_parameters\_experiment1.txt

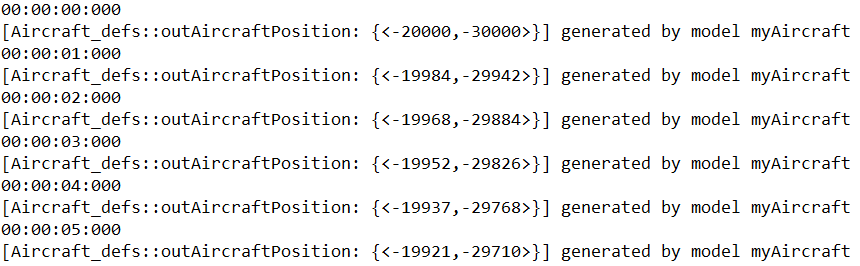
**Output data files:**

* \simulation\_results\radar\_MANPADS\_engagement\_output\_messages\_experiment1.txt
* \simulation\_results\radar\_MANPADS\_engagement\_output\_state\_experiment1.txt

**Discussion:**

1. An Aircraft Position is generated every second:

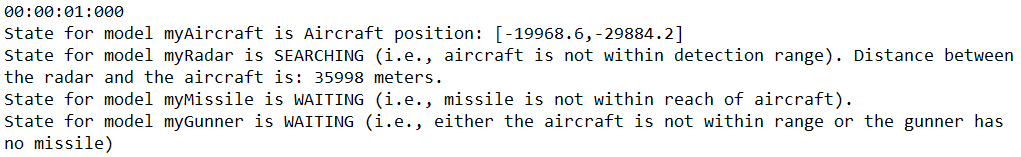
Output message:



[…]

Example of output state:

[…]

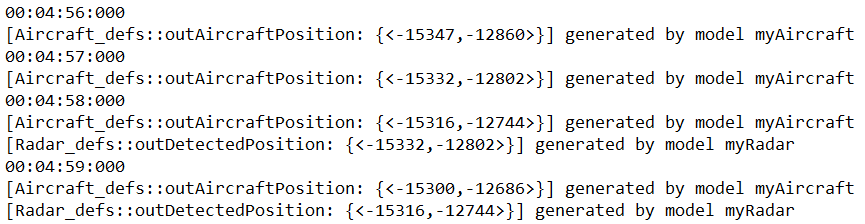


[…]

1. At time 00:04:57, the aircraft is detected by the radar. The radar starts outputting a detected aircraft position every second.

Output message:

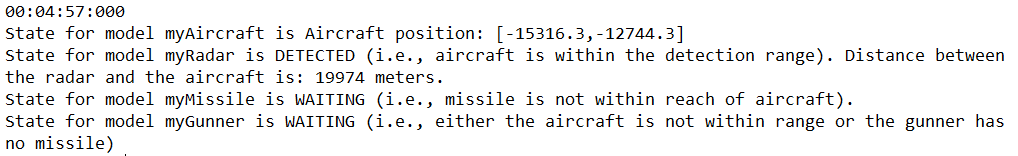
[…]



[…]

Output state:

[…]

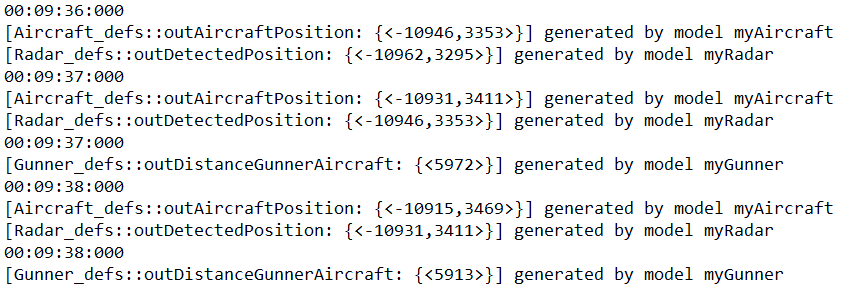


[…]

1. At time 00:09:37, the aircraft is within the missile launch range of the gunner. The gunner model outputs the distance between the gunner and launches a missile.

Output message:

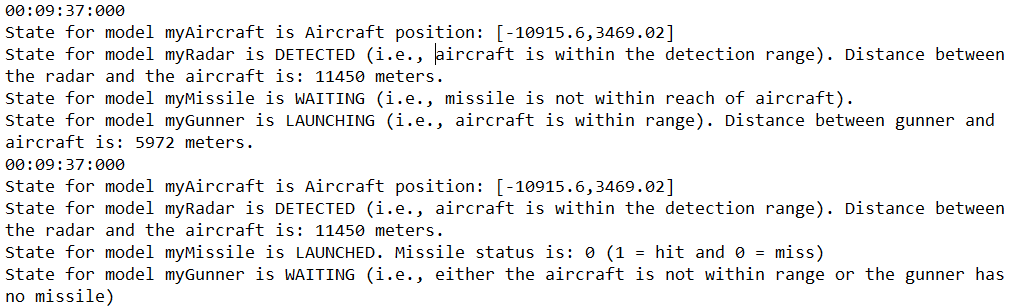
[…]



[…]

Output state:

[…]

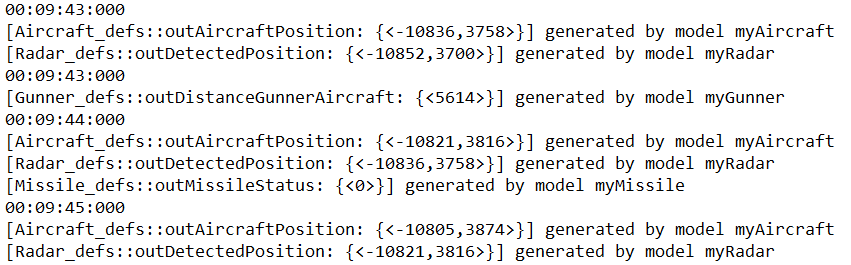


[…]

1. At time 00:09:44, the missile returns a *miss* status, and the gunner starts reloading a new missile. The 7 seconds delay between the previous event and this output has been implemented to represent the missile flying time.

Output message:

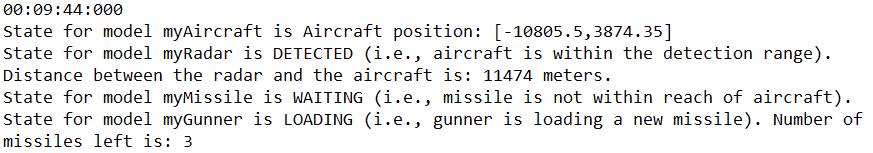
[…]



[…]

Output state:

[…]

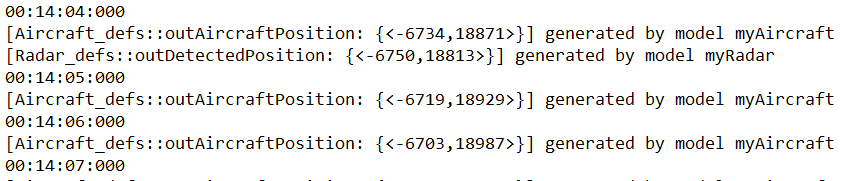


[…]

1. 40 seconds later, the gunner attempts to launch another missile. The missile returns a miss, and the gunner starts reloading. This sequence is repeated until the gunner no longer has missiles.
2. At 00:14:04, the aircraft is no longer within the radar detection range.

Output message:

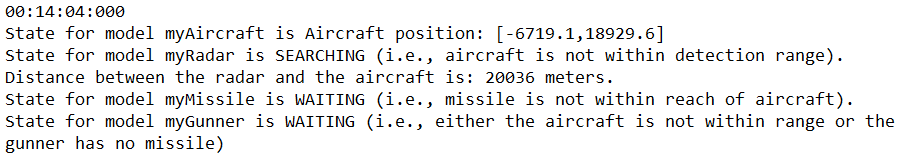
[…]



[…]

Output state:

[…]

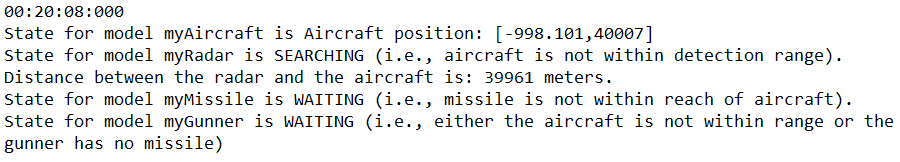


[…]

1. At 00:20:08, the simulation is terminated since the aircraft has reached the end of its predefined flying path (i.e., [-1 000, 40 000].

Output state:

[…]



[…]

### Integration Test #2

The second experiment uses the same mission scenario as experiment #1. This time, the gunner is well experimented. The probability of hit for all ranges has been set to 80%.

**Input data file:**

* \input\_data\engagement\_parameters\_experiment2.txt

**Output data files:**

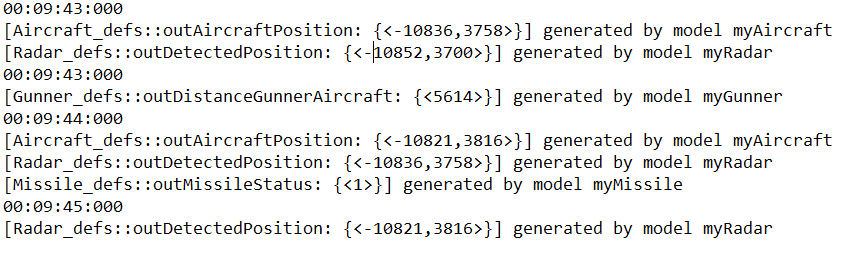
* \simulation\_results\radar\_MANPADS\_engagement\_output\_messages\_experiment2.txt
* \simulation\_results\radar\_MANPADS\_engagement\_output\_state\_experiment2.txt

**Discussion:**

1. As for the previous test, the radar starts detecting the aircraft at time 00:04:57, and at time 00:09:37, the aircraft is within the missile launch range of the gunner.
2. In this test, the aircraft is hit at the first attempt. A missile status of 1 (hit) is returned to the gunner and the aircraft at time 00:09:44. The simulation ends.

Output message:

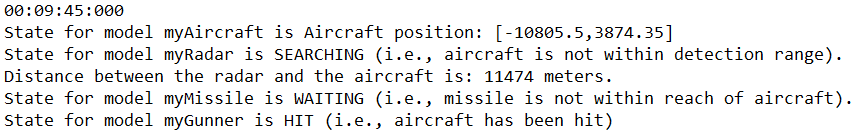
[…]



[…]

Output state:

[…]



[…]

# PART 4 – DEVS WEB VIEWER

The results of integration test #1 were loaded into the DEVS Web Viewer. Compared to the simulation results logs, the animated visualization of the simulation results quickly allows assessing the main events of the scenario. For example, the results of integration test #1 in the DEVS Web Viewer clearly show when the radar is detecting the aircraft and how many missiles have been launched during the simulation. Figure 5 presents a screenshot of the animated visualization of the simulation results using the DEVS Web Viewer.

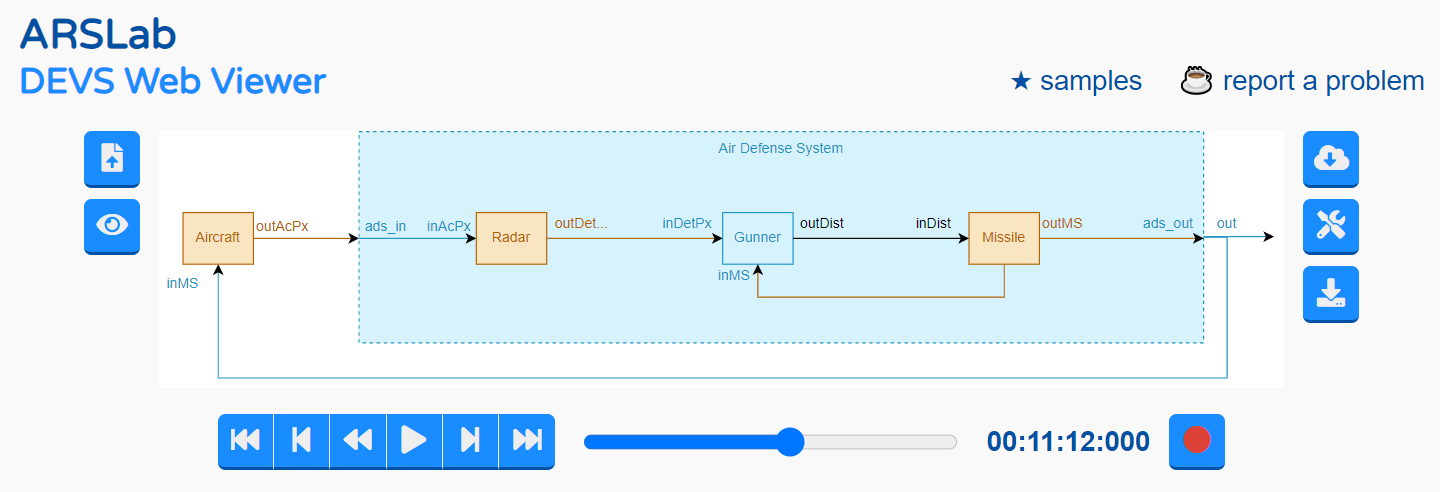


Figure : Screenshot of an animated visualization of the simulation results using DEVS Web Viewer

# PART 5 – FUTURE WORK

The above work was a first attempt to represent a radar-augmented MANPADS engagement using DEVS formalism. Future work to better represent such a system should include the following:

* **Better representation of the MANPADS gunner —** For example, the gunner could include an acquisition time parameter that would vary based on other parameters. This acquisition time could also be associated with the MANPADS’ battery life (typically 60 seconds). If the acquisition time takes longer than 60 seconds, the gunner would have to change the battery and add some delay. The gunner has a limited number of batteries.
* **Integrate more than on MANPADS gunners —** Radar-augmented MANPADS scenarios typically include more than one gunner per radar.
* **Integrate more than one aircraft in flying formation**

1. https://www.defenseworld.net/news/18530/Israel\_s\_Elta\_Wins\_USAF\_Contract\_For\_21\_MANPADS\_Anti\_Aircraft\_Kits [↑](#footnote-ref-1)