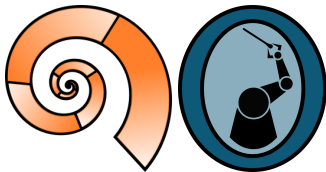


Introduction to the Hybrid Systems module

of the Systems Design Laboratory course

Luca Geretti

University of Verona, Italy





- The focus is on the analysis of systems whose state is defined not only by a valuation on discrete variables, but also on **continuous** variables, i.e., the system has a **hybrid** nature.
 - ▶ We describe **hybrid automata** as a modeling formalism and introduce the concept of their **reachability**



- The focus is on the analysis of systems whose state is defined not only by a valuation on discrete variables, but also on **continuous** variables, i.e., the system has a **hybrid** nature.
 - ▶ We describe **hybrid automata** as a modeling formalism and introduce the concept of their **reachability**
- Due to the complexity of the reachability problem, we distinguish between **static** and **dynamic** analysis of a system.
 - ▶ For each of the two categories we discuss specific aspects and use a software to experiment with the topic



-
1. Introduction to the analysis of hybrid systems
 2. Hybrid automata and the reachability problem
 3. A survey of tools and the rigorous numerical approach
 4. Representations of reachable sets
 5. Bounded reachability
 6. Unbounded reachability
 7. Open systems and modularity
 8. A specific domain: human-robot interaction
 9. A specific application: collision prediction



Many real systems have a double nature. They:

- evolve with a given **continuous** law
- such law may change after specific **discrete** events



Such systems are called **hybrid systems** because they mix **discrete** and **continuous** behaviours

Engineering example: 4-strokes engine



- **Intake stroke:** air and vaporized fuel are drawn in
- **Compression stroke:** fuel vapor and air are compressed and ignited
- **Combustion stroke:** fuel combusts and piston is pushed downwards
- **Exhaust/Emission stroke:** exhaust is driven out
- During the 1st, 2nd and 4th strokes the piston is relying on the power and momentum generated by the pistons of the other cylinders

During the 4 strokes pressure, temperature, . . . , vary continuously

Biology example: escherichia coli



Escherichia coli is a bacterium detecting the food concentration through a set of receptors.

It responds in one of two ways:

- **Directed motion**: moves in a straight line by moving its flagella counterclockwise
- **Tumbling**: randomly changes its heading by moving its flagella clockwise

In either case, each variable changes **continuously** (or it is constant). Discontinuities in the direction of the flagella are handled by changing between the two **discrete** receptor responses.

The cost of errors and failures



- Often hybrid systems represent **safety-critical systems**:
 - ▶ airplane control systems, medical care systems, train signalling systems, automotive systems, ...
- Bugs, design errors and failures can cause catastrophic loss of money, time or even human life:



- ▶ Ariane 5 explosion (1996, approx \$500 million)



- ▶ Therac 25 accident (1985-87, six patients seriously injured or killed)



- The usual methods for analyzing the behavior of a hybrid system are:
 - ▶ **Testing** (using the system itself)
 - ▶ **Simulation** (using a model of the system)



- The usual methods for analyzing the behavior of a hybrid system are:
 - ▶ **Testing** (using the system itself)
 - ▶ **Simulation** (using a model of the system)
- They can cover only a subset of possible behaviors of the system



- The usual methods for analyzing the behavior of a hybrid system are:
 - ▶ **Testing** (using the system itself)
 - ▶ **Simulation** (using a model of the system)
- They can cover only a subset of possible behaviors of the system
- They can only prove the existence of errors, not their in-existence



- The usual methods for analyzing the behavior of a hybrid system are:
 - ▶ **Testing** (using the system itself)
 - ▶ **Simulation** (using a model of the system)
- They can cover only a subset of possible behaviors of the system
- They can only prove the existence of errors, not their in-existence
- Hybrid systems are **reactive systems**: they maintain an ongoing interaction with the environment
 - ▶ errors and failures caused by the interaction with the environment can be very difficult to discover!



To provide guarantees on the results of analysis, it is necessary to resort to **formal methods**.

- Rigorous models with well-defined semantics
- Set-based computation instead of (multiple) point-based
- Proper termination guarantees in order not to miss any behaviors



To provide guarantees on the results of analysis, it is necessary to resort to **formal methods**.

- Rigorous models with well-defined semantics
- Set-based computation instead of (multiple) point-based
- Proper termination guarantees in order not to miss any behaviors

A rigorous approach requires a significantly **larger computational cost**, which scales with the dimension of the system worse than non-rigorous approaches.

Static vs dynamic analysis



Static (a.k.a. offline or design-time)

In this “conventional” approach we analyze a model of the system in **isolation** with respect to the system modeled.

Static vs dynamic analysis



Static (a.k.a. offline or design-time)

In this “conventional” approach we analyze a model of the system in **isolation** with respect to the system modeled.

- The model may be arbitrarily inaccurate

Static vs dynamic analysis



Static (a.k.a. offline or design-time)

In this “conventional” approach we analyze a model of the system in **isolation** with respect to the system modeled.

- The model may be arbitrarily inaccurate
- There is no time budget on the analysis

Static vs dynamic analysis



Static (a.k.a. offline or design-time)

In this “conventional” approach we analyze a model of the system in **isolation** with respect to the system modeled.

- The model may be arbitrarily inaccurate
- There is no time budget on the analysis

Dynamic (a.k.a. online or run-time)

In this alternative approach there is a real time **interaction** between the running model of the system and the actual system.

Static vs dynamic analysis



Static (a.k.a. offline or design-time)

In this “conventional” approach we analyze a model of the system in **isolation** with respect to the system modeled.

- The model may be arbitrarily inaccurate
- There is no time budget on the analysis

Dynamic (a.k.a. online or run-time)

In this alternative approach there is a real time **interaction** between the running model of the system and the actual system.

- Model evolution and system evolution can be compared, allowing model refinement

Static vs dynamic analysis



Static (a.k.a. offline or design-time)

In this “conventional” approach we analyze a model of the system in **isolation** with respect to the system modeled.

- The model may be arbitrarily inaccurate
- There is no time budget on the analysis

Dynamic (a.k.a. online or run-time)

In this alternative approach there is a real time **interaction** between the running model of the system and the actual system.

- Model evolution and system evolution can be compared, allowing model refinement
- The analysis is expected to be performed periodically while the system evolves, hence a time budget is enforced



Static analysis: Ariadne



Library for rigorous numerical analysis and synthesis of hybrid automata.

- Open source C++ with Python bindings
- Runs on macOS and Linux

Tools we will use in the module



Static analysis: Ariadne



Library for rigorous numerical analysis and synthesis of hybrid automata.

- Open source C++ with Python bindings
- Runs on macOS and Linux

Dynamic analysis: Opera



Runtime for analysis and control of interaction between humans and robots.

- Open source C++
- Runs on macOS, Linux and Windows