Computational Models for Embedded Systems

Vescan Andreea, PHD, Assoc. Prof.

Faculty of Mathematics and Computer Science
Babeș-Bolyai University
Cluj-Napoca

2025-2026

Lecture 13a: Hybrid Systems

Software Systems Verification and Validation

"Tell me and I forget, teach me and I may remember, involve me and I learn."

(Benjamin Franklin)

Outline

- Hybrid systems
- Hybrid Dynamical Models
- Designing Hybrid Systems
  - Automated Guided Vehicle
  - Obstacle Avoidance with Multi-robot Coordination

- Next lecture:
  - Dynamical systems (today)

Hybrid systems

- Hybrid Dynamical Models

- Hybrid systems admit more general forms of continuous-time evolution for state variables using differential and algebraic equations.
- Such models provide a unified framework for designing and analyzing systems that integrate computation, communication, and control of the physical world.

- The most commonly used commercial design tool for such systems is Stateflow/Simulink
  - Stateflow is used to describe discretely updated variables using state machines.
  - Simulink is used to describe continuous-time dynamical systems.

Hybrid systems

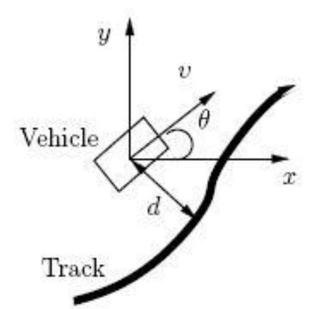
• Hybrid Dynamical Models (cont)

- The model of computation for hybrid processes is a generalization of the model for timed processes.
- A hybrid process is described using an **extended state machine** with **modes** and **switches**.
    - Each process has input, output, and state variables, and some of these variables are of type cont. A variable of type cont takes values from the set of real numbers (or an interval of real numbers), and is updated continuously as time progresses while a process waits in a mode.
    - A mode switch is executed discretely, and takes zero time. As usual, such a switch is guarded with a condition over state and input variables, can update state and output variables, and describes either an input, or an output, or an internal action.
    - A mode is annotated with differential and algebraic equations that specify how state and output variables of type cont evolve. In addition, each mode also specifies a constraint on how long the process can wait in that mode using a Boolean expression over state variables.

Hybrid systems
- Designing Hybrid Systems
oAutomated Guided Vehicle

- Consider an autonomous vehicle that needs to be programmed to move along a track as closely following the track as possible. The track is not known to the vehicle in advance, but is equipped with sensors. In particular, assuming that the vehicle has not strayed too far from the track, the sensors can provide the distance d of the vehicle from the center of the track. Such information can be provided, for instance, by placing photodiodes along the track.
- The vehicle dynamics is modeled as a planar rigid-body motion with two degrees of freedom. It can move forward along its body axis with speed v, where the speed must be between 0 and 10 mph, and it can rotate about its center of gravity with an angular speed omega, which can range over the interval from -pi to + pi radians/second. If the coordinates (x; y) model the position of the vehicle, and gives the relative angle (with respect to some fixed planar global frame) in which the vehicle is headed, then its motion is described by the equations

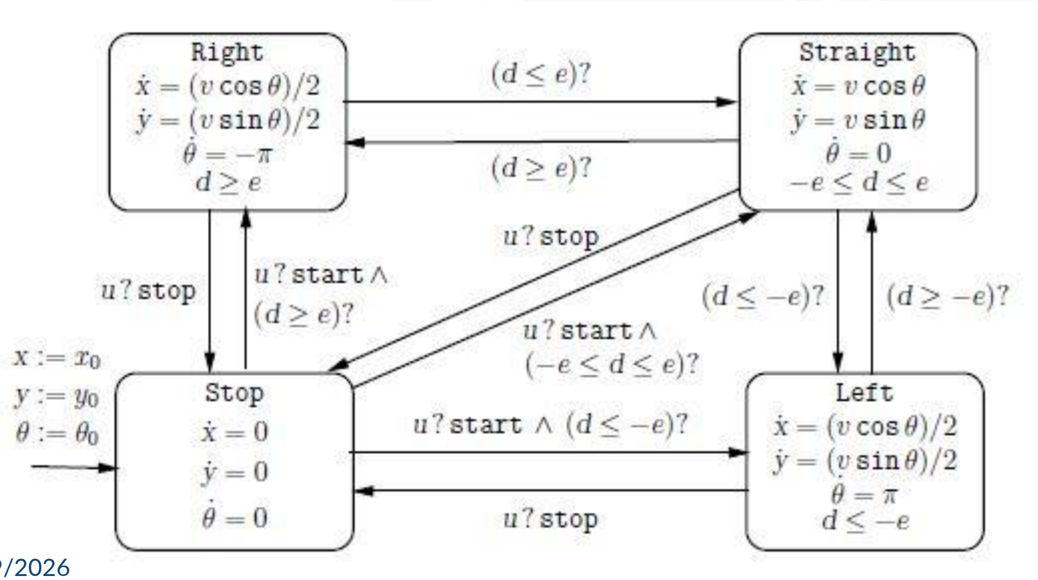
$$\dot{x} = v\cos\theta; \quad \dot{y} = v\sin\theta; \quad \dot{\theta} = \omega.$$

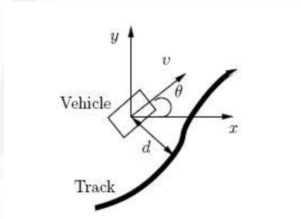


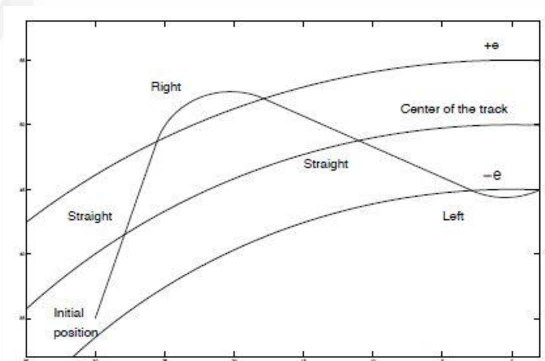
Hybrid systems
- Designing Hybrid Systems
oAutomated Guided Vehicle (cont)

•Based on the current measurement of the distance d, the controller must adjust the control inputs v and *omega* so as to keep the value of the distance d as close to 0 as possible. The control design problem is additionally constrained by the requirement that the vehicle hardware provides only three settings for the angular speed *omega*: it can be 0, or +pi, or -pi.

•The control designer makes a further design decision that when the vehicle is headed straight, it moves as fast as possible, with speed v equal to 10 mph. When the vehicle is turning left or turning right, it attempts to do so at half the maximum possible speed, namely 5 mph. This leads to 4 modes of operation as shown in the hybrid state machine.







Hybrid systems
- Designing Hybrid Systems
o Obstacle Avoidance with Multi-robot Coordination

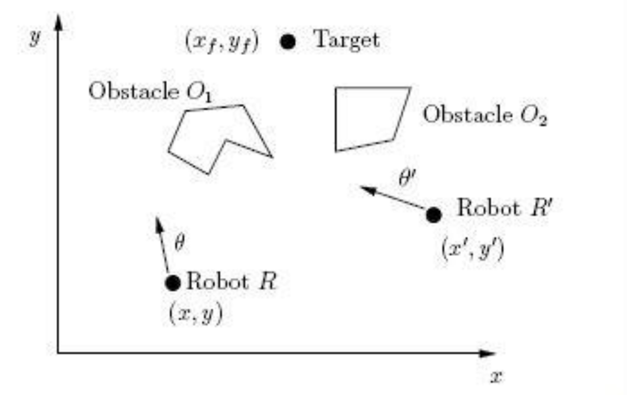
- A challenging application domain for modeling and analysis of hybrid systems is design of multi-robot coordination for a system of **autonomous mobile robots**. A typical surveillance task involves identifying a target, and exploring a room with unknown geometry, possibly with obstacles, to reach the target.
- The sensory capabilities of each robot yield only imperfect information about the surroundings, and in particular, each robot has only estimates about the obstacle positions.
- The robots can send information to one another over wireless links, and use this information to improve the accuracy of their estimates for better path planning.

Hybrid systems
- Designing Hybrid Systems

oObstacle Avoidance with Multi-robot Coordination (cont)

- A specific scenario

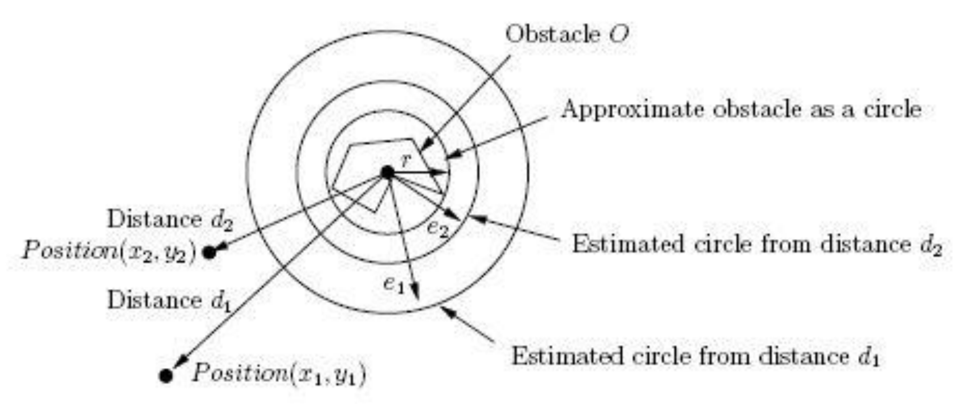
•To illustrate modeling in a concrete scenario, suppose there are two autonomous mobile robots R and R'. We assume a two-dimensional world in which each robot is modeled as a point in the two-dimensional X-Y plane.



Hybrid systems
- Designing Hybrid Systems
oObstacle Avoidance with Multi-robot Coordination (cont)

• Estimating obstacles

•Mapping obstacles accurately using images from a camera is a computationally expensive task. Furthermore, optimal path planning to a target location given complex descriptions of obstacles is also computationally expensive. To address these difficulties, let us estimate each obstacle using a circle.

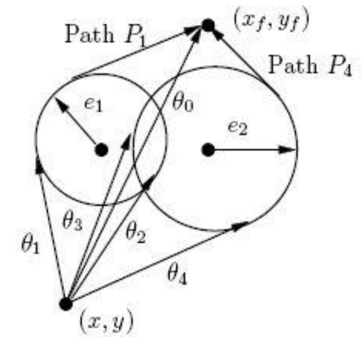


Hybrid systems
- Designing Hybrid Systems

- Obstacle Avoidance with Multi-robot Coordination (cont)
  - Path planning

Consider the robot R at position (x, y). Its goal is to reach the target (x_f, y_f) while avoiding the two obstacles. For the obstacle O_1, the area it occupies according to the robot R is a circle with center (x_o^1, y_o^1) with the current estimated radius e_1. Similarly, for the obstacle O_2, the area it occupies according to the robot R is a circle with center (x_o^2, y_o^2) with the current estimated radius e_2. The objective of the planner is to compute a shortest path from the current position to the target so that the trajectory does not intersect the estimated obstacle circles.

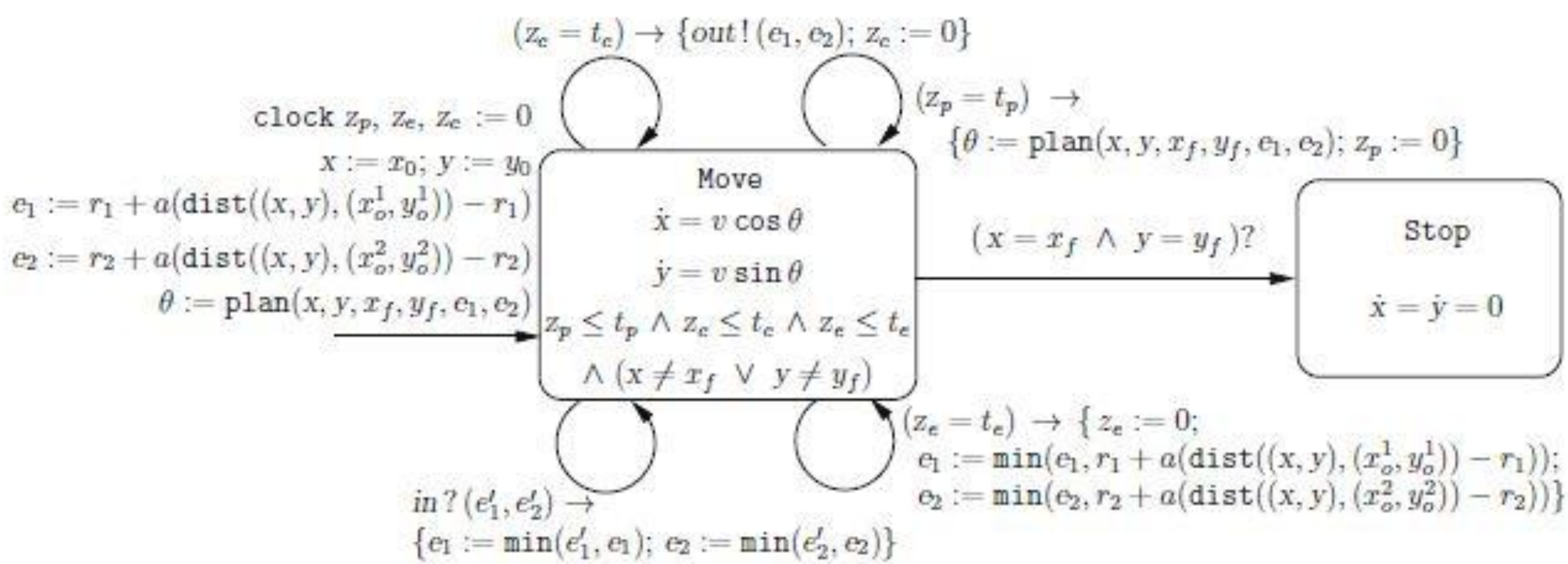
The plan is usually updated in a discrete manner. In our example, the planning algorithm is invoked every t_p seconds, and the planning algorithm determines the control input θ that gives the direction for the robot motion. The direction stays unchanged till the next time the planning algorithm is invoked. We assume that the planning algorithm is captured by the function `plan` that takes as input the current position (x, y), the target (x_f, y_f), the first obstacle circle given by the center (x_o^1, y_o^1) and radius e_1, and the second obstacle circle given by the center (x_o^2, y_o^2) and radius e_2, and returns the direction θ in which the robot should head.



- A specific scenario

Hybrid systems
- Designing Hybrid Systems

- Obstacle Avoidance with Multi-robot Coordination (cont)
  - Hybrid modeling



The hybrid state machine for the robot

References

Sources

[1] Rajeev Alur, Principles of Cyber-Physical Systems, MIT Press, 2015

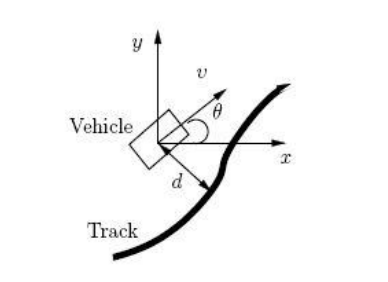
(Biblioteca de Matematica si Informatica, UBB,

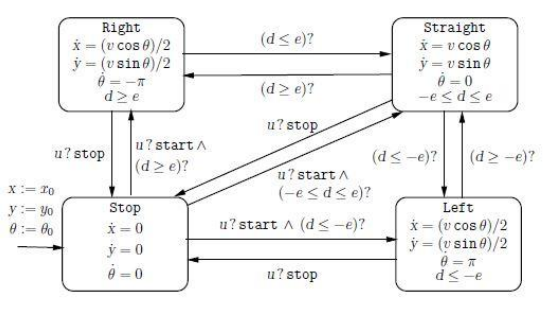
- Rajeev Alur, Principles of Cyber-Physical Systems, 2015
  COTA:9491
- https://www.bcucluj.ro/ro/despre-noi/filiala/biblioteca-de-matematic%C4%83-%C5%9Fi-informatic%C4%83)

CMES – Today Bring it All Together

Hybrid Systems

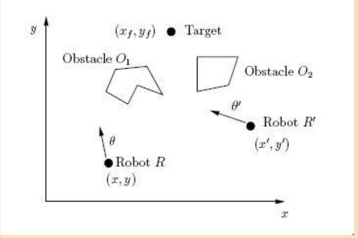
- Automated Guided Vehicle

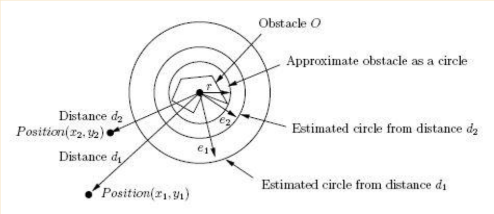


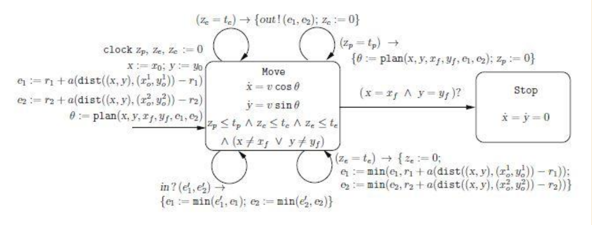


Hybrid Systems

- Obstacle Avoidance with Multi-robot Coordination







Next Lecture (today)

- Dynamical Systems

Software Systems Verification and Validation

“Tell me and I forget, teach me and I may remember, involve me and I learn.”

(Benjamin Franklin)