Mission control of autonomous vehicles based on time logic framework

Danilo F. Bassi

Outline

- Introduction:
  - Autonomous vehicle
  - Multilevel Control
  - Mission Control
- Formulation of logic paradigm for mission control
- Synthesis of a dynamic controller
- Application of mission controller
- Conclusions

Introduction

Autonomous vehicle (mobile robot)

- Vehicle: mobility
  - UGV
  - UAV
  - etc.
- Autonomy: minimize or eliminate human participation in the control
- Purpose: accomplish a desired task or mission
- Problem: provide full autonomous control given an unknown and unstructured environment

Introduction

Control of autonomous vehicles

- Multilevel control
  - Hierarchical approach: a divide-to-conquer method for solving complex problem in structured way.
  - Higher levels have an abstraction of lower
  - Commands are given to lower level to be translated into more detailed actions
- Three level control for robotic systems:
  - Mission level: solves a task (highest level)
  - Navigation level: solves a movement
  - Locomotion level: produces motion

Introduction

Mission specification and control

- Objective: to satisfy requirements or commands from the user, given *some* assumptions
- High level of abstraction
- It is a control: specification is given (reference) and outcome is always verified (feedback)
- Solution:
  - Find the set of actions that system will perform to satisfy the requirements
  - Select the best action to accomplish the mission

Formulation of logic paradigm

Logic control paradigm

- Logic programming: use mathematical logic for computer programming
- Declarative paradigm: goals are stated and the system find the right actions
- It is close to natural language: easy and precise to formalize goals
- Solution can be shown to be correct by construction (theorem proving)
- Controller is a logical consequence of the symbolic model of the system combined with the control specification

Formulation of logic paradigm

Logic formalism for mission control

- The logic formalism can capture the nature of mission control: initial conditions, goals.
- Natural human interface: a natural language statement is formalied as logical formula.
- Discrete: the mission problem is transformed into a discrete control problem.
- Use of modal logic for dealing with time dependency of dynamic systems.
- A transition system: a finite state machine is used for control.

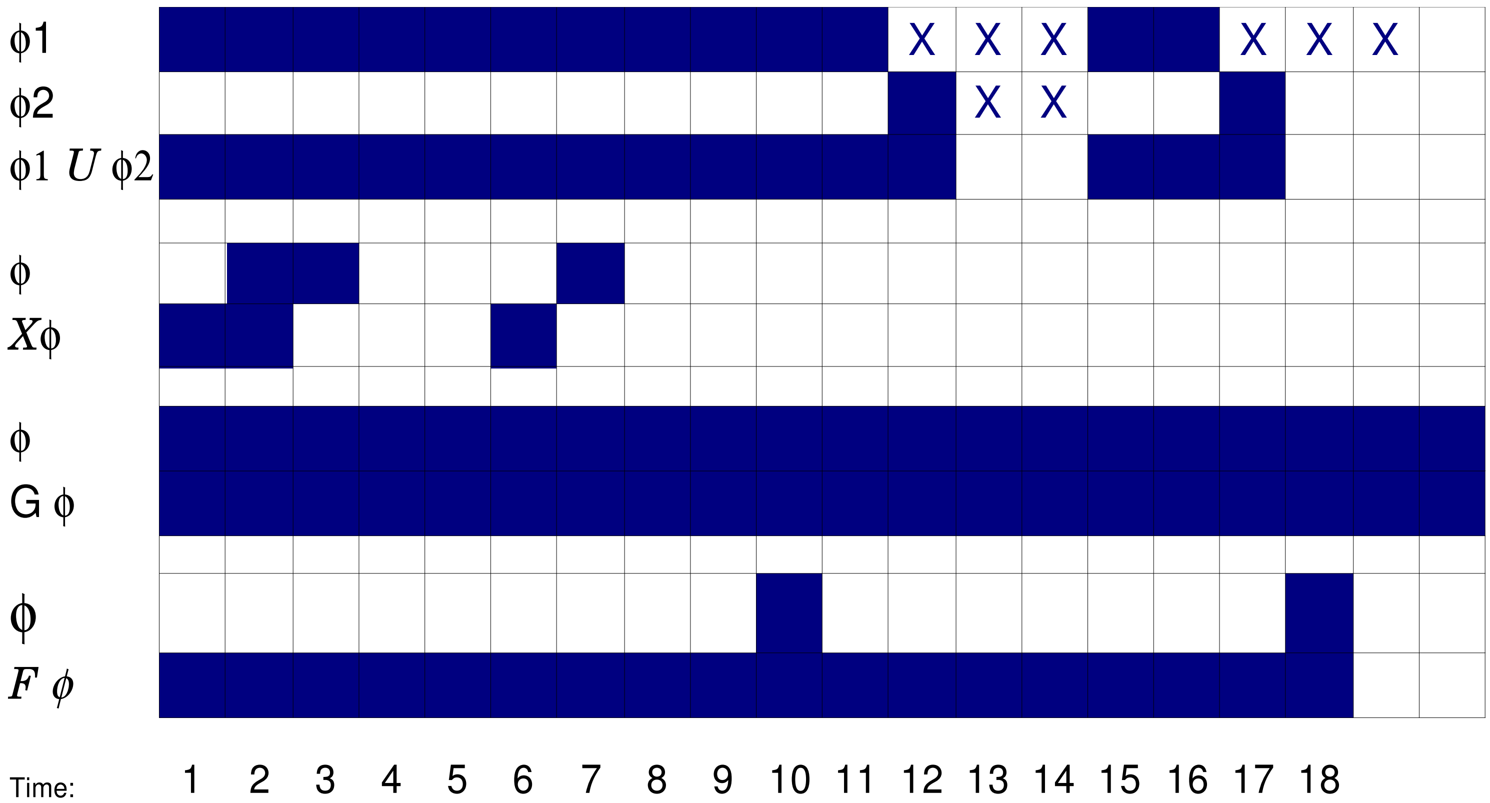
Formulation of logic paradigm

Temporal Logic

- Temporal logic have been proposed as the specification language (linear temporal logic)
- System of logic that deals with modalities that qualify the truth in term of time
- New operators are added to the traditional Boolean logic:
  - Until: *U*
  - Next: *X* *(neXt)*
  - Always: *G (Globally)*
  - Eventually: *F (Future)*

Formulation of logic paradigm

Temporal Logic time diagrams



Formulation of logic paradigm

Definitions

- Abstract model: a simplified symbolic model of a dynamic system. Sufficient details to define desired missions of the system:
  - Symbolic states: Q
  - Symbolic inputs: U
  - Transition function: δ
  - Finite State Machine (FSM): formed by previous elements
- Atomic propositions: are logic assertions about the state of the system (true or false), a function of the state.

Formulation of logic paradigm

Definitions

- Discrete transition system: M=$(Q, Q_0, U, \delta, \Pi, \Vdash)$, with Q the set of states, $Q_0 \subseteq Q$ set of initial states, U the set of system inputs ,δ:QxU $\mapsto 2^Q$ the transition function, Π set of atomic propositions and $\Vdash \subseteq Q \times \Pi$ the satisfaction relation.
- Run: a run is sequence of states that satisfy the transition relation: $q^0, q^1, q^2 \ldots$ with $q^i \in Q$, $q^0 \in Q_0$, $q^{i+1} \in \delta(q^i, u^i)$ $\forall$ $i \geq 1$ and $u^i \in U$
- Syntax: A LTL formula $\phi \in L_{LTL}(\Pi)$, belongs to the linear temporal language defined over Π that satisfies the grammar: $\phi ::= p \mid \neg\phi \mid \phi \vee \phi \mid \phi U \phi \mid X \phi$, with ϕ formula and $p \in \Pi$ proposition , $\neg$ negation, $\vee$ or, U until, X next

Formulation of logic paradigm

Linear Temporal Logic (LTL) Semantics

- The semantics gives the meaning of a formula: it is defined over a *run* σ that satisfies the formula ϕ at position i, denoted by $\sigma(i) \vDash \phi$.
- The semantics is defined as follows:

$\sigma(i) \vDash p$ iff $(q^i, p) \in \Vdash$ (or $q^i \Vdash p$)

$\sigma(i) \vDash \neg\phi$ iff $\sigma(i) \nvDash \phi$

$\sigma(i) \vDash \phi_1 \vee \phi_2$ iff $\sigma(i) \vDash \phi_1$ or $\sigma(i) \vDash \phi_2$

$\sigma(i) \vDash \phi_1 U \phi_2$ iff $\exists j \geq i$ such that $\sigma(j) \vDash \phi_2$ and $\sigma(k) \vDash \phi_1$, $\forall k \in [i, j-1]$

$\sigma(i) \vDash X\phi$ .. iff $\sigma(i+1) \vDash \phi$

Where p is an atomic proposition, and ϕ, ϕ_1,ϕ_2 are formulas

- A run σ satisfies a formula ϕ, $\sigma \vDash p$, if it does at the initial position of the sequence: $\sigma(0) \vDash p$,

Formulation of logic paradigm

Mission Control Logic Formulation

- Given a Discrete Transition System $(Q, Q_0, U, \delta, \Pi, \Vdash)$,
- Specification: a mission is specified as a LTL formula ϕ defined over Π (set of atomic propositions): $\phi \in L_{LTL}(\Pi)$
- Feasible solutions: are the runs that satisfy the formula ϕ, the set $\sum_{\phi} = \{\sigma \mid \sigma \vDash \phi\}$

  where $\sigma = q^0 q^1 \ldots q^i \in Q,\ q^0 \in Q_0,\ q^{i+1} \delta, (u^i, q^i)\ \ \forall\ i \geq 1,\ u^i \in U$

Development of logic paradigm

Mission Control formulation steps

- Identify the real dynamic system and its environment:
  and find an abstraction (FSM)
- Find a set of primitive propositions : Π. This set will be the base for the specification language.
- Identify the discrete transition function δ
- Formulate the desired mission specification as a LTL formula ϕ defined over Π
- Solve the corresponding feasibility or optimality discrete control problem: find a control sequence u that produces a state sequence $q^0 q^1 \ldots q^i$ which satisfies the given formula

Development of logic paradigm

Examples of formulas for mission control

- Goal reaching and obstacle avoidance
  - Mission formula $\phi = \Diamond F\ goal \wedge G \neg(\vee_i\ obstacle_i)$.
  - Semantics: eventually (F) reach the destination (goal) and never ($G\neg$: *always not*) go over any obstacle $(\vee_i\ obstacle_i)$.
- Fault tolerance and recovery
  - Mission formula $\phi = FG\ (\Psi \vee (\Psi\ U\ \ (\phi_f \wedge X\rho_f))$
  - Semantics: ψ normal operation, ϕ_f is the occurrence of a specific fault f, ρ_f recovery procedure for fault f.
- Target seeking and following
  - Mission formula $\phi = G(\psi\ U\ (\theta\ \wedge X\rho))$
  - Semantics: always do seeking a target (ψ), until one is found $(\theta\ )$ and then continue following that target (ρ)

Synthesis of a mission controller

Mission control solutions for LTL formulation

- Main problem
  - Find a control sequence given a LTL formula specification
- Issues
  - Is it possible to satisfy the formula?
  - How to select a solution (best one?)
- Approaches:
  - Static
  - Dynamic

Synthesis of a mission controller

Static approaches

- Most of actually known solutions for control synthesis from LTL formula specification are static and fixed.
- The appeal of the static approach: full solution can be found beforehand and that the feasibility can be checked
- Typically two automata are defined: one for verifying the acceptance of the formula (Büchi automaton) and the controller, after the first automaton is combined with the FSM model.
- Issues of static approaches:
  - High complexity of Büchi automaton synthesis (exponential space with respect to formula size).
  - The solution es essencially fixed, whereas the robotic environment is dynamic and variable

Synthesis of a mission controller

Formulation of Dynamic approach

- The solution need not be computed in full, but can be found in a more progressive manner.
- This approach follows the general framework of the dynamic programming
  - The solution is built from an actual (present) state.
  - Once a partial solution is found, it is stored, so there is no repetition previous calculations.
  - Optimality principle is used: namely for any optimal problem, each part of the problem needs to be optimal.
- Drawback: no guarantee of finding a solution (or it may need to be improved), may require to do look ahead search

Synthesis of a mission controller

Developed dynamic approach (sketch)

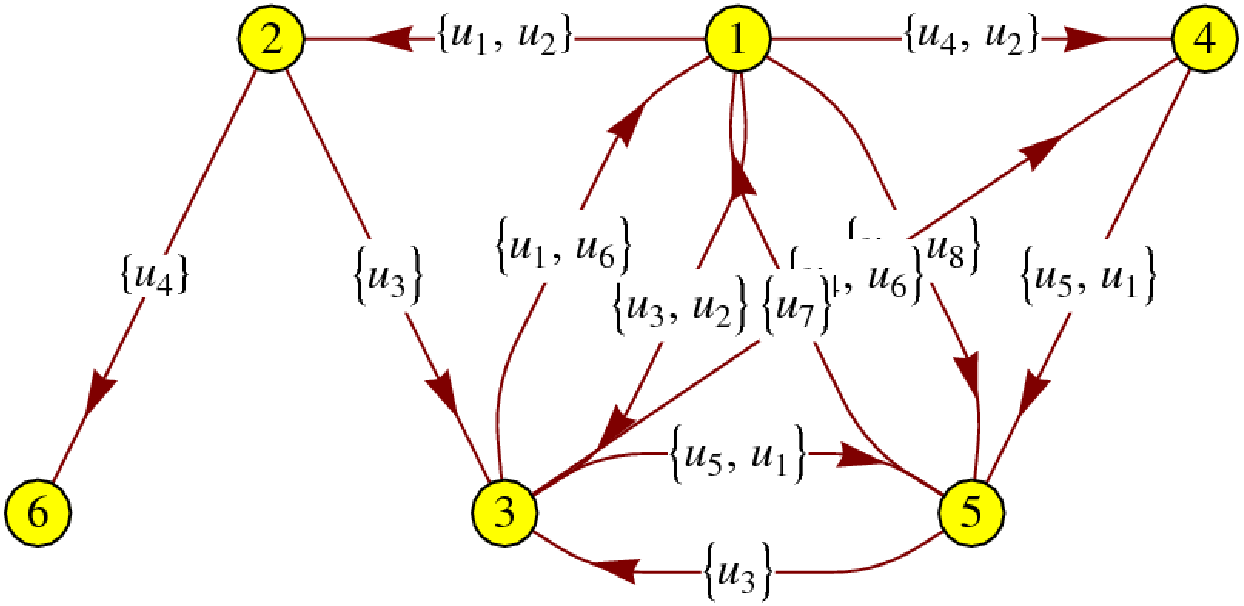
- Definition of control states as a pairs (q,ϕ): system states and TLT formula
- Definition of control transitions $(qi,\phi i)\rightarrow(qi+1,\phi i+1)$: combines system transitions and formulaor successors
- Finding resolution states (q,True) or (q,False): in these states the control automaton will halt
- The control problem is reduced to finding a path between an actual state and a resolution state.
- For finite satisfaction, the end state must be (q,True)
- The resolution paths are stored
- The required input (u) is deduced from transition function δ

Application of mission controller

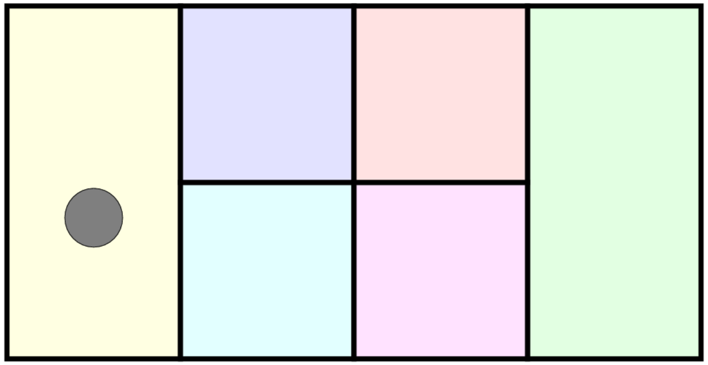
- Mission control over a synthetic discrete system (automaton)
  - Arbitrary automaton and TLT formula
  - Small system: detailed solution can be followed and analysed
- Robotic task solution
  - A specific robotic system: Robotic Cleaner Vehicle

Application of mission controller

- Synthetic discrete system



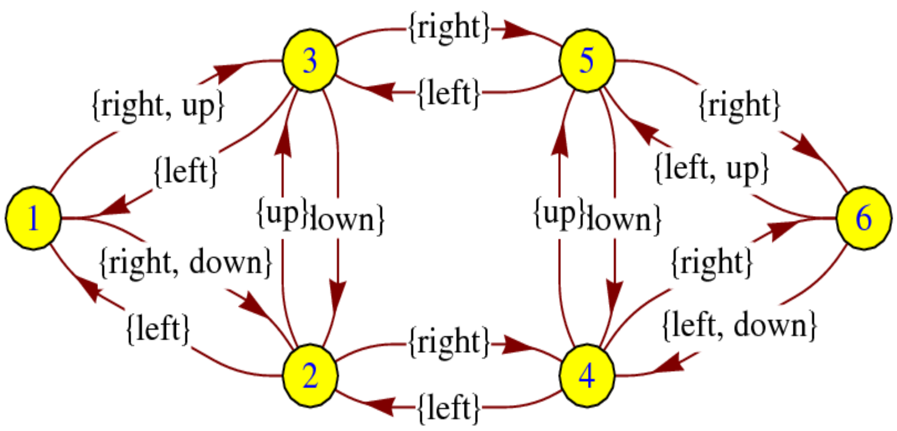
Application of mission controller



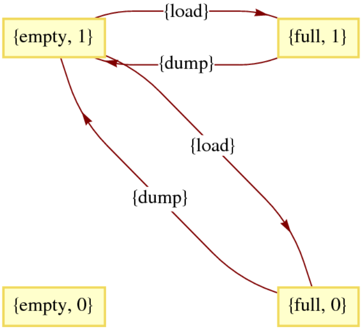
Robotic Cleaner

- Setting: rectangular area, divided in 6 sectors. Each sector has a determined units of “dirt”
- Vehicle: ground robot, with limited environment sensor (clean or dirty), and one unit loading capability (empty or loaded)
- Mission: to clean different sectors, and bring everything in one area. This mission shall be defined as a TLT formula

Application of mission controller



Motion FSM



Action FSM

Definitions

- Motion Commands {right, left, up, down}
- Action Commands {load, dump}
- Logical Propositions {cleaned, loaded, position1, position2,…}
- Finite State Machine (FSM): formed by combining motion FSM and action FSM (product automaton)

Conclusions

- The Logic Paradigm solution to the problem of mission control is based on the satisfability of a linear time logical (LTL) formula over a discrete system (FSM):
  - LTL is very expressive to deal with dynamic and time relations
  - Logic paradigm satisfaction translates a specification into actions
- A novel solution for the LTL formula satisfaction problem was developed.
- The main contribution of this research is a dynamic formulation that progressively builts the required structures and functions while running the machine.

Conclusions

- This is a practical solution, with lower computational complexity that other approaches (static). This is very convenient where they are h/w or process constraints.
- Complex robotic mission can be expressed by the LTL language and then solved by this dynamic approach.
- This powerful paradigm could be also used in many other applications where automatic actions are looked to satisfy given requirements.