Test-Based Induction of Finite-State Machines with Continuous Output Actions

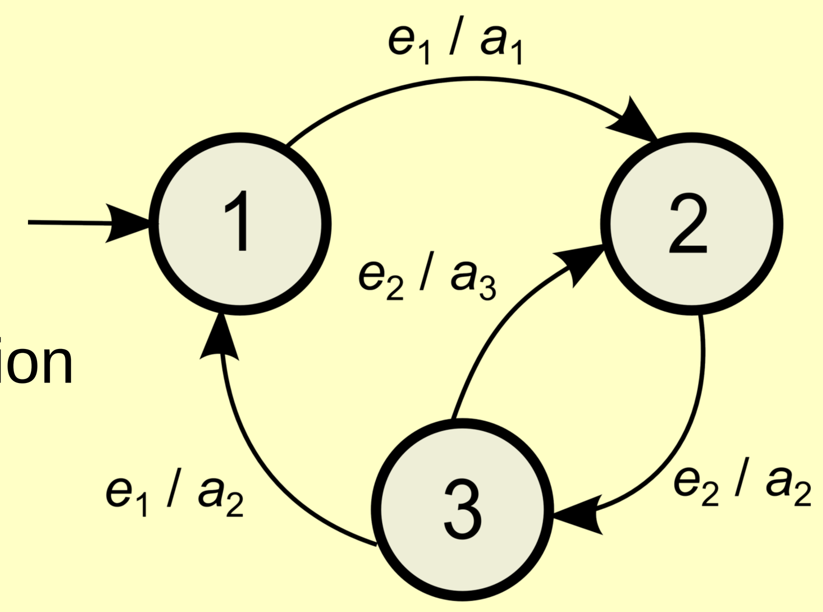
Igor Buzhinsky, Vladimir Ulyantsev, Anatoly Shalyto

St. Petersburg National Research University of Information Technologies, Mechanics and Optics
Computer Technologies Department

Problem Statement

Finite-State Machine:

- FSM = (S, Σ, Δ, δ, λ, s_0)
- S – finite set of states
- Σ, Δ – event and action sets
- δ: $S \times E \rightarrow S$ – transition function
- λ: $S \times E \rightarrow \Delta$ – output function
- s_0 – start state



Problem:

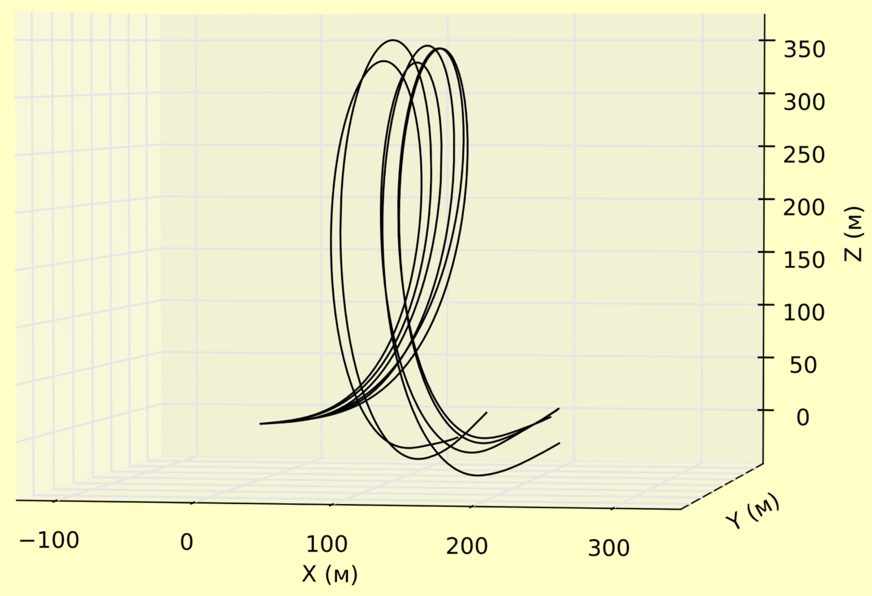
- Control object has continuous (real) control parameters
- Tests are the examples of proper control
- Given a set of N tests (N ≈ 20–30), an FSM should be constructed with behavior close to the tests
- Aircraft model is used as a control object
- Tests can be written manually in a flight simulator

Input Data: Tests

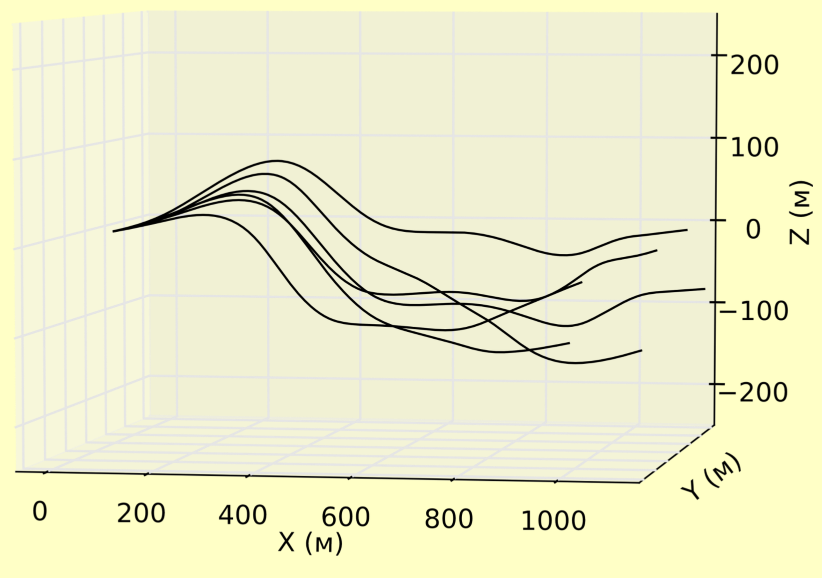
- in[i][t][k] – inputs (flight parameter values)
- out[i][t][j] – outputs (control parameter values)

Values	Meaning	t = 1	...	t = 235
in[i][t][1]	Pitch angle	3.078	...	4.112
in[i][t][2]	Airspeed (knots)	251.42	...	253.20
out[i][t][1]	Aileron position	0.000	...	0.073
out[i][t][2]	Elevator position	-0.035	...	-0.037

Test example

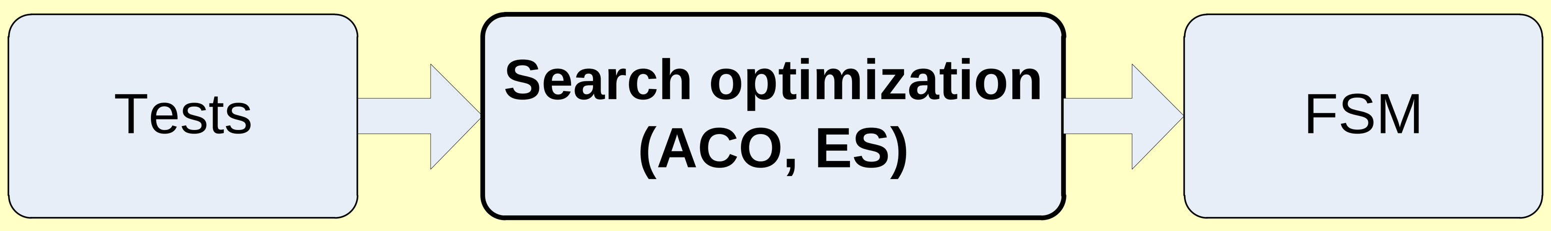


Loop test set (trajectories)

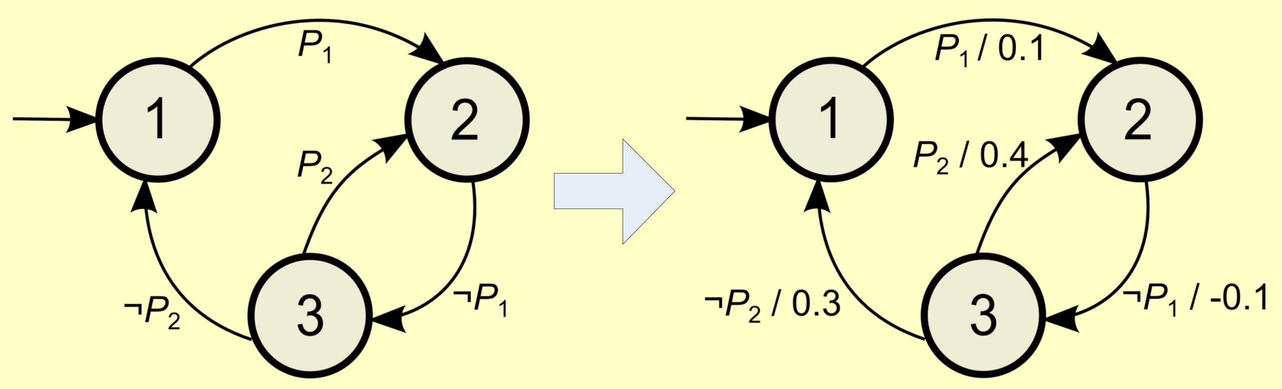


Barrel roll test set (trajectories)

Method Summary

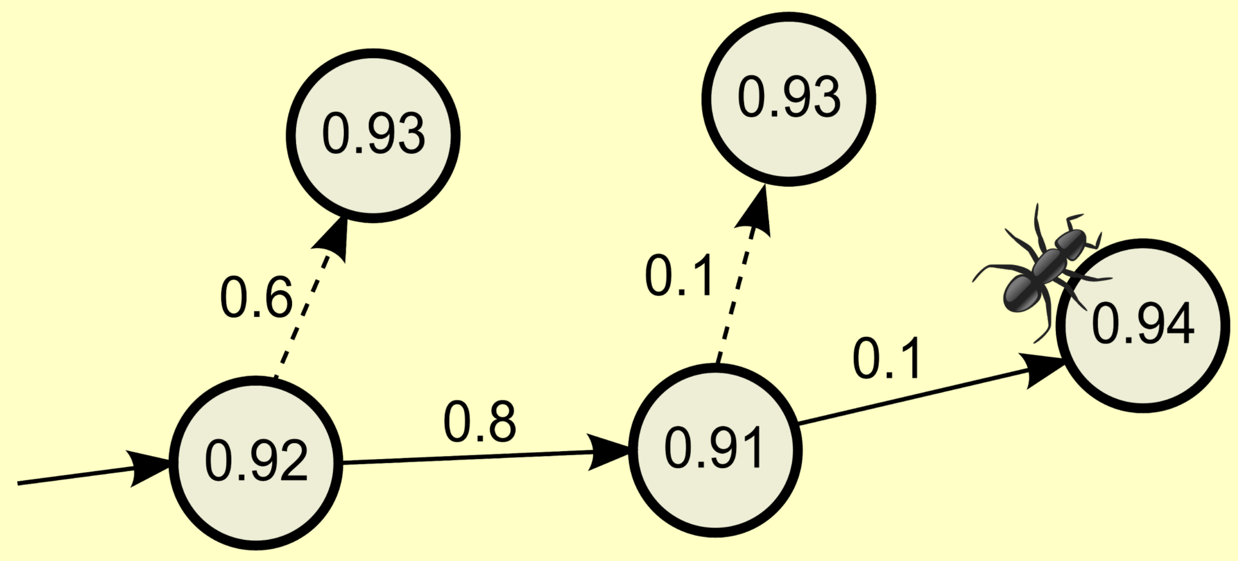


- **Ant colony optimization** (ACO) and **evolution strategy** (ES) are used instead of genetic algorithm (GA) which was used earlier for the same problem
- Predicates transform continuous inputs to discrete events
- FSMs without actions (FSM skeletons) are ACO / ES individuals
- Actions are assigned so that fitness function is maximized:



Ant Colony Optimization

- Construction graph: vertices – FSM skeletons, edges – mutations (small changes in skeletons)
- Initially, graph consists of a single randomly generated vertex / skeleton
- Graph grows during algorithm execution
- Ants wander using *pheromone*



A part of the construction graph

Experiments & Results

- Compared ACO, (μ, λ)-ES and GA
- Intel Core 2 Quad Q9400 processor, four cores
- 25 runs of ACO / ES / GA for each test set
- Searching for FSMs with four states
- Numbers of runs in which the fitness values were reached:

Fitness value	ACO	(μ, λ)-ES	GA
0.9890	11	8	0
0.9887	21	18	2
0.9884	24	24	8
0.9881	24	24	17
0.9878	24	24	21

Results for the loop test set

Fitness value	ACO	(μ, λ)-ES	GA
0.9884	8	3	0
0.9882	23	18	5
0.9880	25	24	15
0.9878	25	24	19
0.9876	25	24	24

Results for the barrel roll test set

- ACO and (μ, λ)-ES outperform GA, ACO slightly outperforms ES
- Run time ≈ 20 minutes
- About 90% of generated FSMs were able to perform the aerobatic figures

Screenshots (FlightGear)

Loop

Barrel roll

Publications

- Ulyantsev V., Tsarev F. Extended Finite-State Machine Induction using SAT-Solver / Proceedings of the 14th IFAC Symposium "Information Control Problems in Manufacturing - INCOM'12". IFAC, 2012, pp. 512-517
- Chivilikhin D., Ulyantsev V. Learning Finite-State Machines with Ant Colony Optimization // Lecture Notes in Computer Science, 2012, Volume 7461/ 2012, pp. 268-275

{buzhinsky, ulyantsev}@rain.ifmo.ru