Search-Based Construction of Finite-State Machines with Real-Valued Actions: New Representation Model

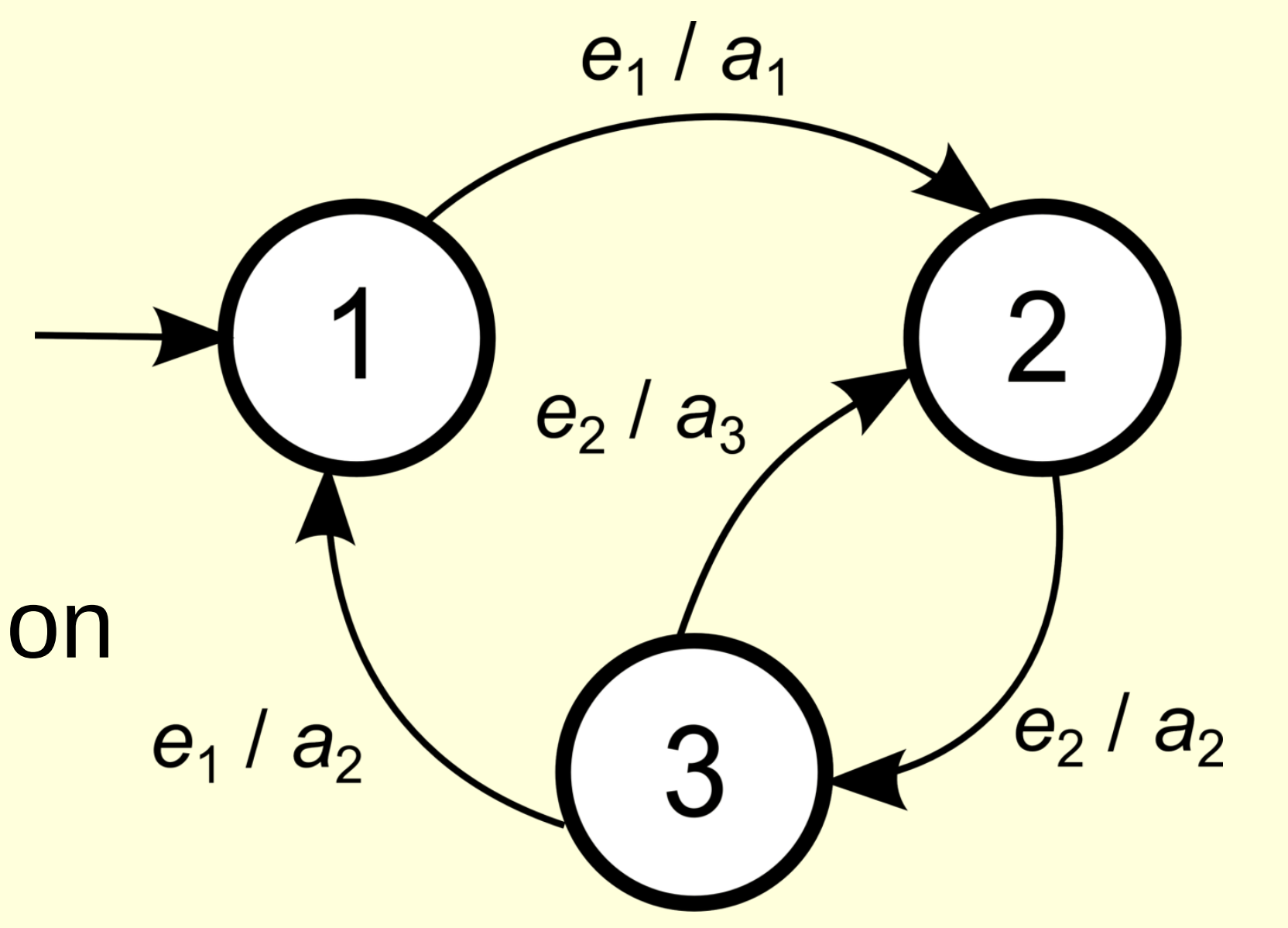
Igor Buzhinsky, Vladimir Ulyantsev, Fedor Tsarev, Anatoly Shalyto

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

Problem Statement

Finite-State Machine:

- FSM = (S, s_0, E, A, δ, λ)
- S – finite set of states
- s_0 – start state
- E, A – event and action sets
- δ: S × E → S – transition function
- λ: S × E → A – output function



Problem:

- Control object has real-valued inputs and 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

Tests (input data):

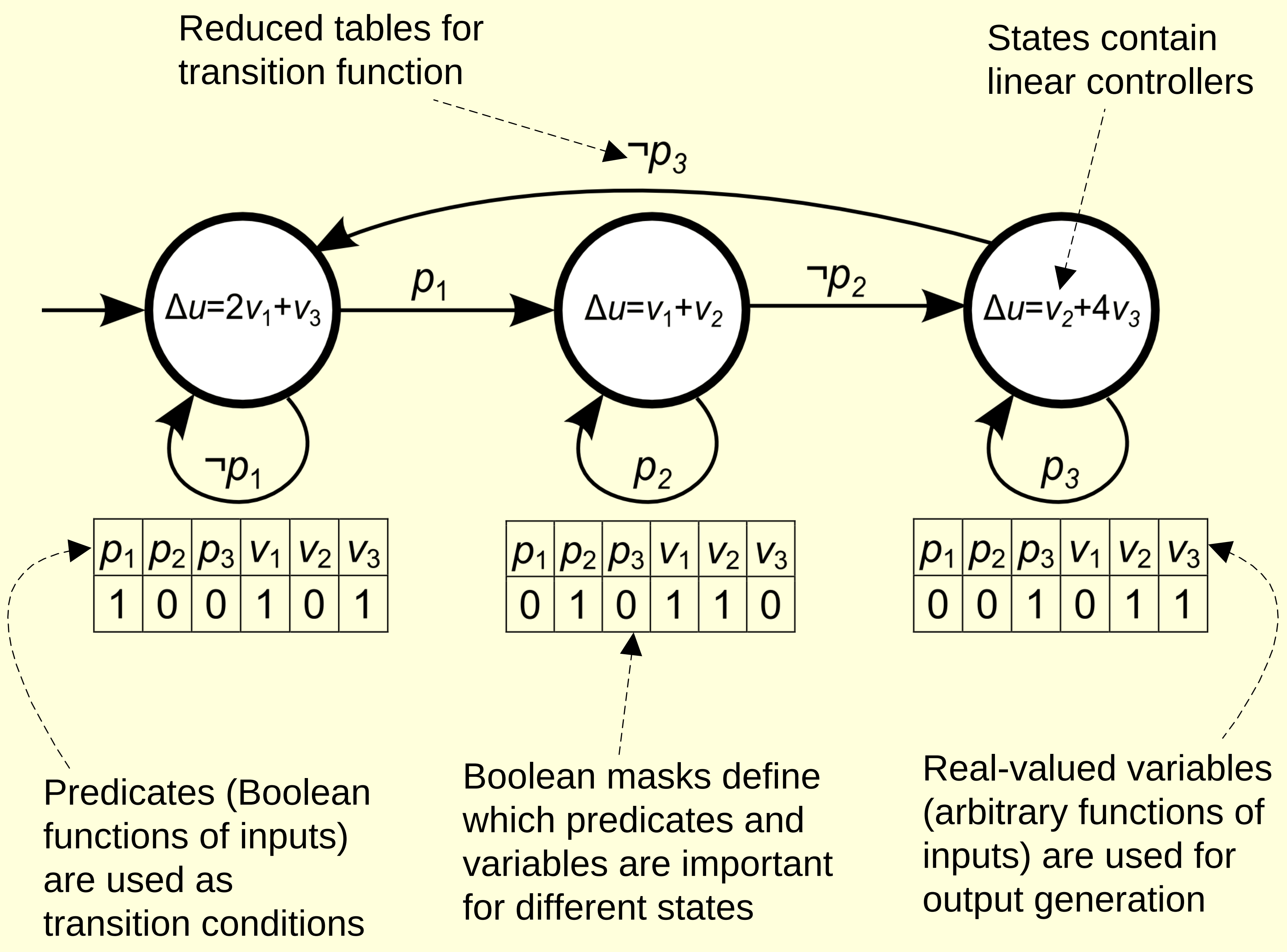
- in[i][t][k] – inputs (flight parameter values)
- out[i][t][j] – outputs (control parameter values)

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

- Aircraft model is used as a control object
- Tests can be written manually in a flight simulator

FSM Representation



p_1	p_2	p_3	v_1	v_2	v_3
1	0	0	1	0	1

p_1	p_2	p_3	v_1	v_2	v_3
0	1	0	1	1	0

p_1	p_2	p_3	v_1	v_2	v_3
0	0	1	0	1	1

Reduced tables for transition function

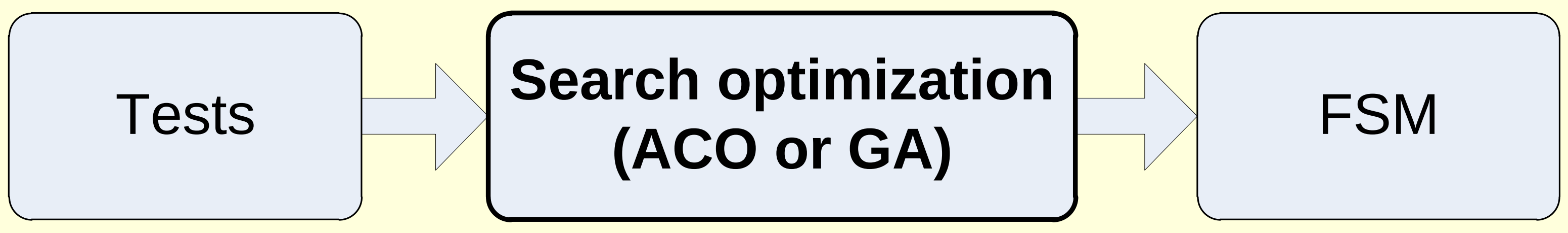
States contain linear controllers

Predicates (Boolean functions of inputs) are used as transition conditions

Boolean masks define which predicates and variables are important for different states

Real-valued variables (arbitrary functions of inputs) are used for output generation

FSM Construction



- **Ant colony optimization** (ACO) and **genetic algorithm** (GA) are used for FSM searching
- FSMs with the undefined output function are individuals
- Output function is derived so that the fitness function is maximized
- Fitness function:

$$f = 1 - \sqrt{\frac{1}{N}\sum_{i=1}^{N}\rho^2(\mathrm{ans}[i],\mathrm{out}[i])} - K\sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\max(\tau_i-\tau,0)\right)^2};$$

$$\rho^2(\mathrm{ans}[i],\mathrm{out}[i]) = \frac{1}{\mathrm{len}[i]}\sum_{t=1}^{\mathrm{len}[i]}\frac{1}{C}\sum_{j=1}^{C}\left(\frac{\mathrm{ans}[i][t][j]-\mathrm{out}[i][t][j]}{c_j^{\max}-c_j^{\min}}\right)^2$$

- C – number of control parameters
- ans[i] – FSM's output for the i-th test
- τ_i – number of state changes on the i-th test, τ – number of allowed state changes

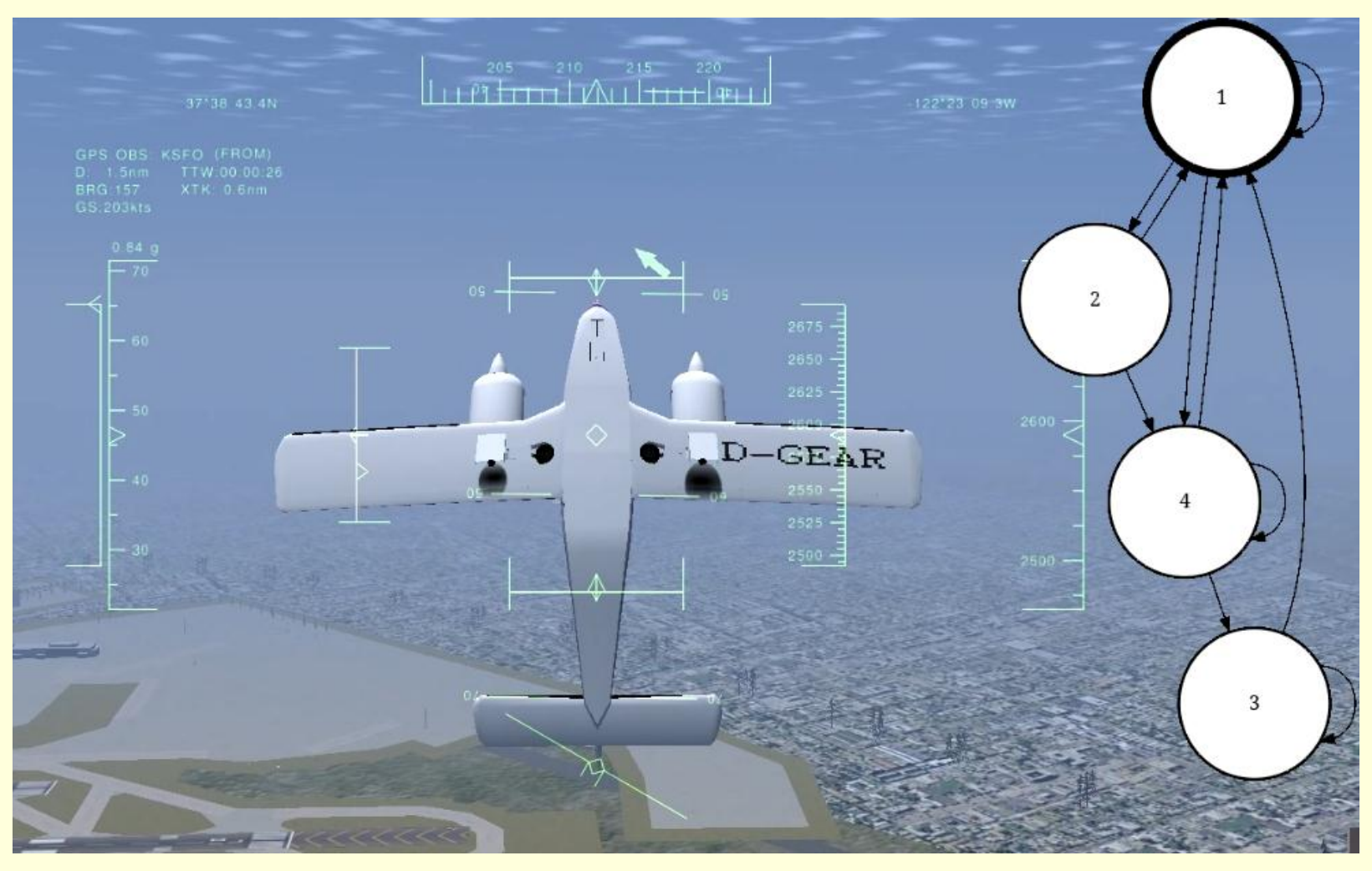
Experiments & Results

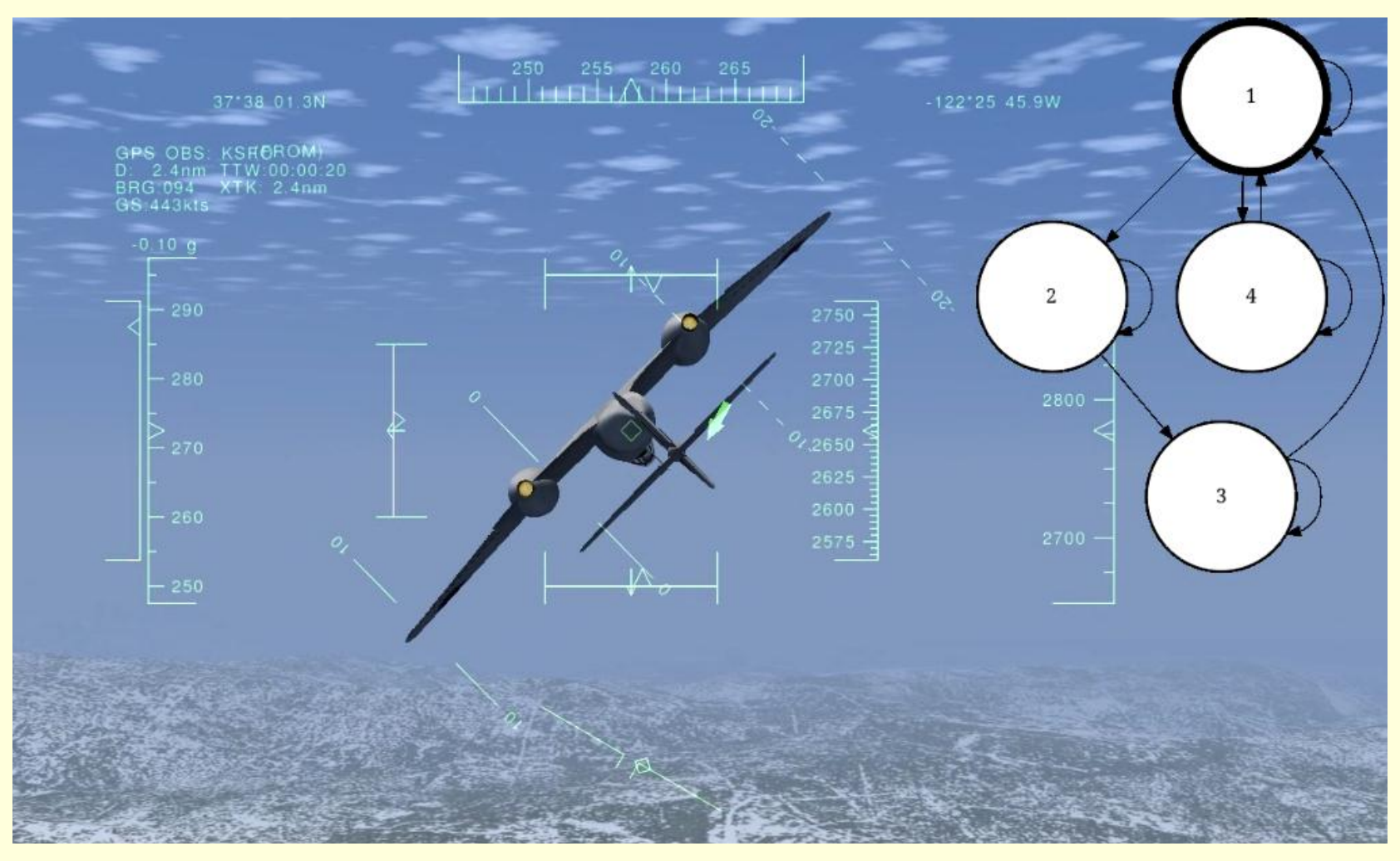
- Intel Core 2 Quad Q9400 processor, four cores
- Three test sets, searching for FSMs with 3–5 states
- Comparison with the previous model (four states):

		Average fitness (ACO)	Average pitch error	Average roll error
Loop	New representation model	0.9866	13.8881	2.1673
	Model from Alexandrov et al.	0.9834	18.0996	5.1842
Barrel roll	New representation model	0.9862	2.2089	15.0424
	Model from Alexandrov et al.	0.9854	4.4626	21.6019
180° turn	New representation model	0.9899	1.6847	3.183
	Model from Alexandrov et al.	0.9900	8.8672	54.1725

- Quality is improved
- Now it is possible to construct FSMs performing the turn
- Method run time ≈ 20 minutes

Screenshots (FlightGear)


Loop


Barrel roll

Publications

- Alexandrov A., Sergushichev A., Kazakov S., Tsarev F. Genetic algorithm for induction of finite automata with continuous and discrete output actions. In Proceedings of the 13th annual conference companion on Genetic and evolutionary computation (GECCO '11), 2011, P. 775–778
- Chivilikhin D., Ulyantsev V. Learning Finite-State Machines with Ant Colony Optimization. Lecture Notes in Computer Science, 2012, Volume 7461/2012, P. 268–275

{buzhinsky, ulyantsev, tsarev}@rain.ifmo.ru