Recognising situations in a flight simulator environment

GAME-ON 2002 Conference

Patrick Ehlert, Quint Mouthaan and Leon Rothkrantz

November 30, 2002

TU Delft
Delft University of Technology

Overview of presentation

- The ICE project
- FlightGear simulator
- Explorative data analysis
- Knowledge based approach
- Conclusions and results
- Future work

The ICE project

- Intelligent Cockpit Environment (ICE)

- Problem:

  Increased automation can result in reduced pilot situation awareness and information overload

- Solution:

  Pilot's assistant, intelligent interface

The ICE project

- Ultimate goal:

  Create system to experiment with intelligent pilot-vehicle interface

- Subgoals:
  - Situation recogniser (SR)
  - Pilot workload assessor
  - Interface decision-maker

The ICE project

- Purpose of SR:

  determine current situation using sensor data from:

  - aircraft (altitude, height, airspeed etc.)
  - pilot (moving stick, throttle etc.)
  - flight plan (expected situations and actions)

- SR can be used as first step to A.I. pilot bot

FlightGear simulator

- Reasons for using FlightGear
  - Open source
  - Multi-platform
  - Extendable
  - Realistic (in most situations)
  - Multiple planes and flight dynamics models
  - XML parameter files
  - User friendly (mailing-list support)

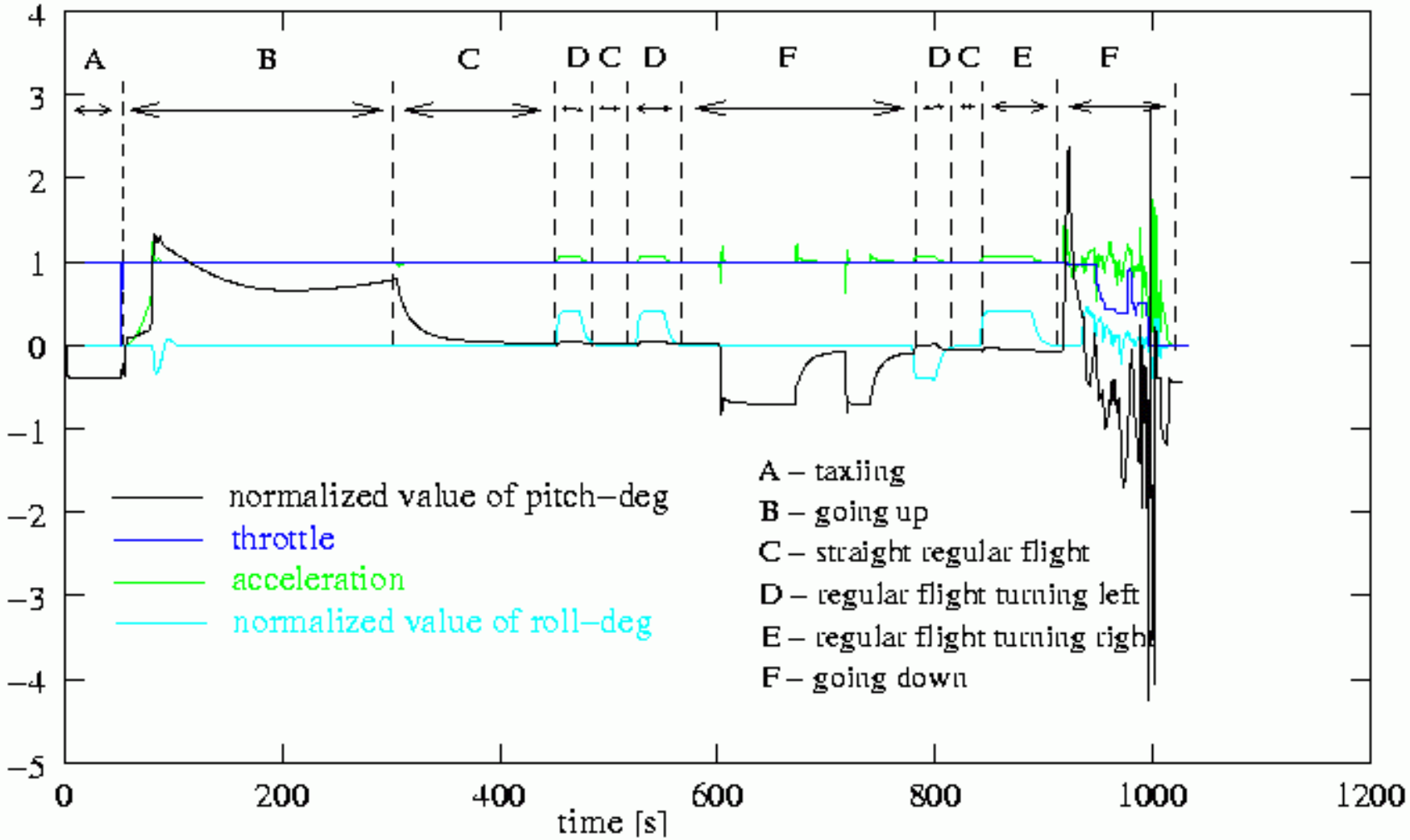
FlightGear simulator

elft

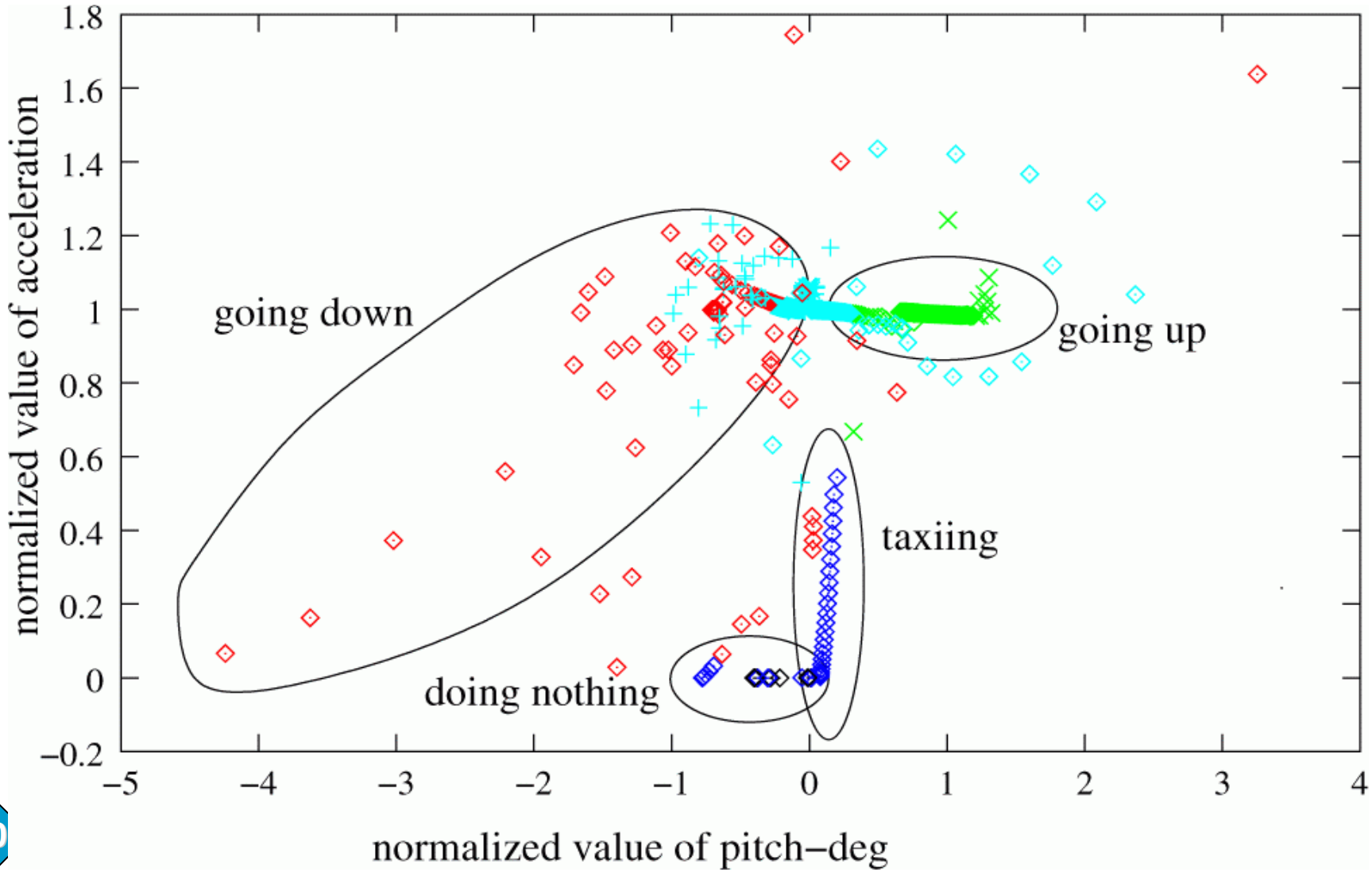
Explorative data analysis

- PCA analysis
  - Clustering of data into states
- Elman neural network
  - Automatically recognise states
  - Predict future states

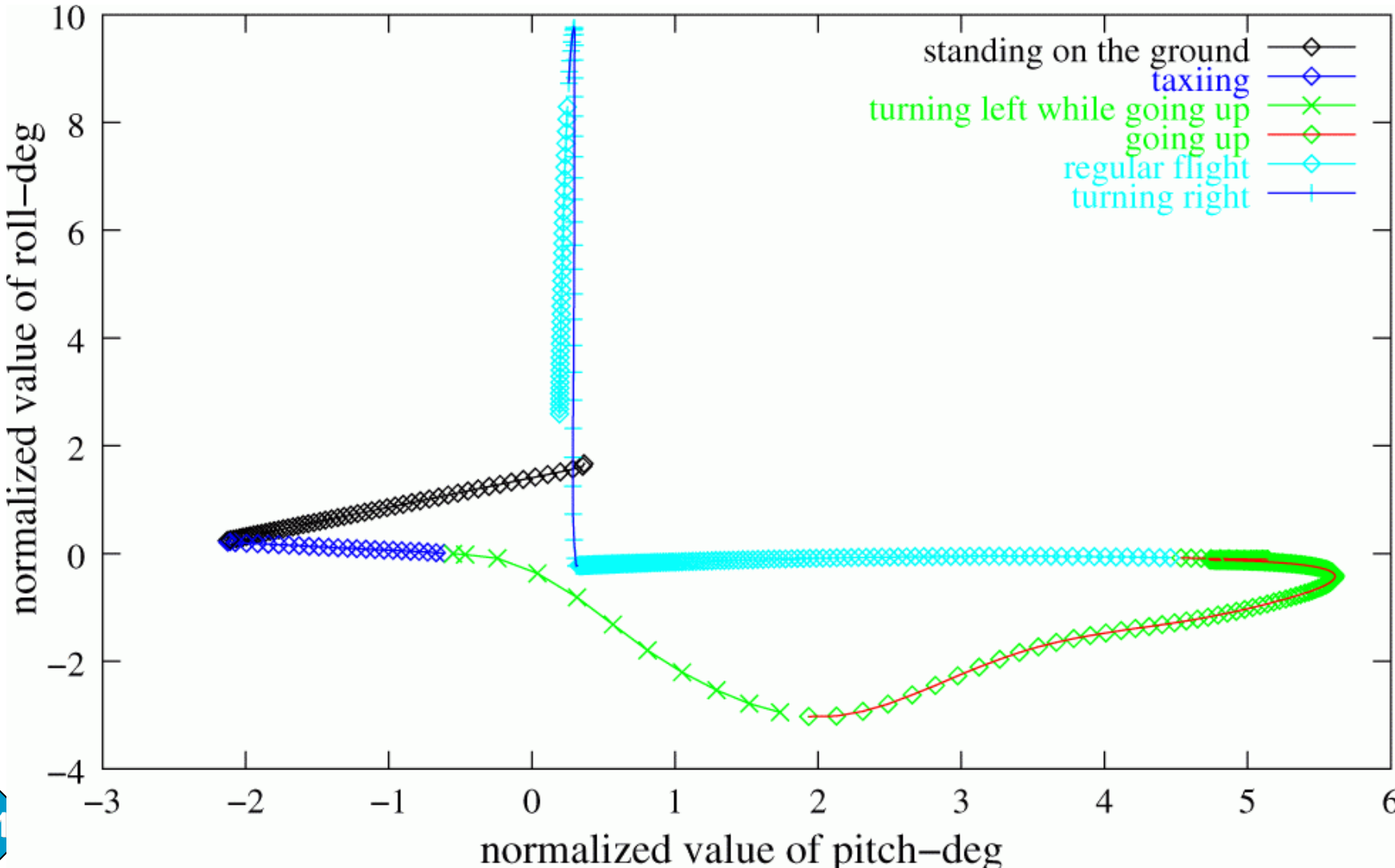
Experiment data



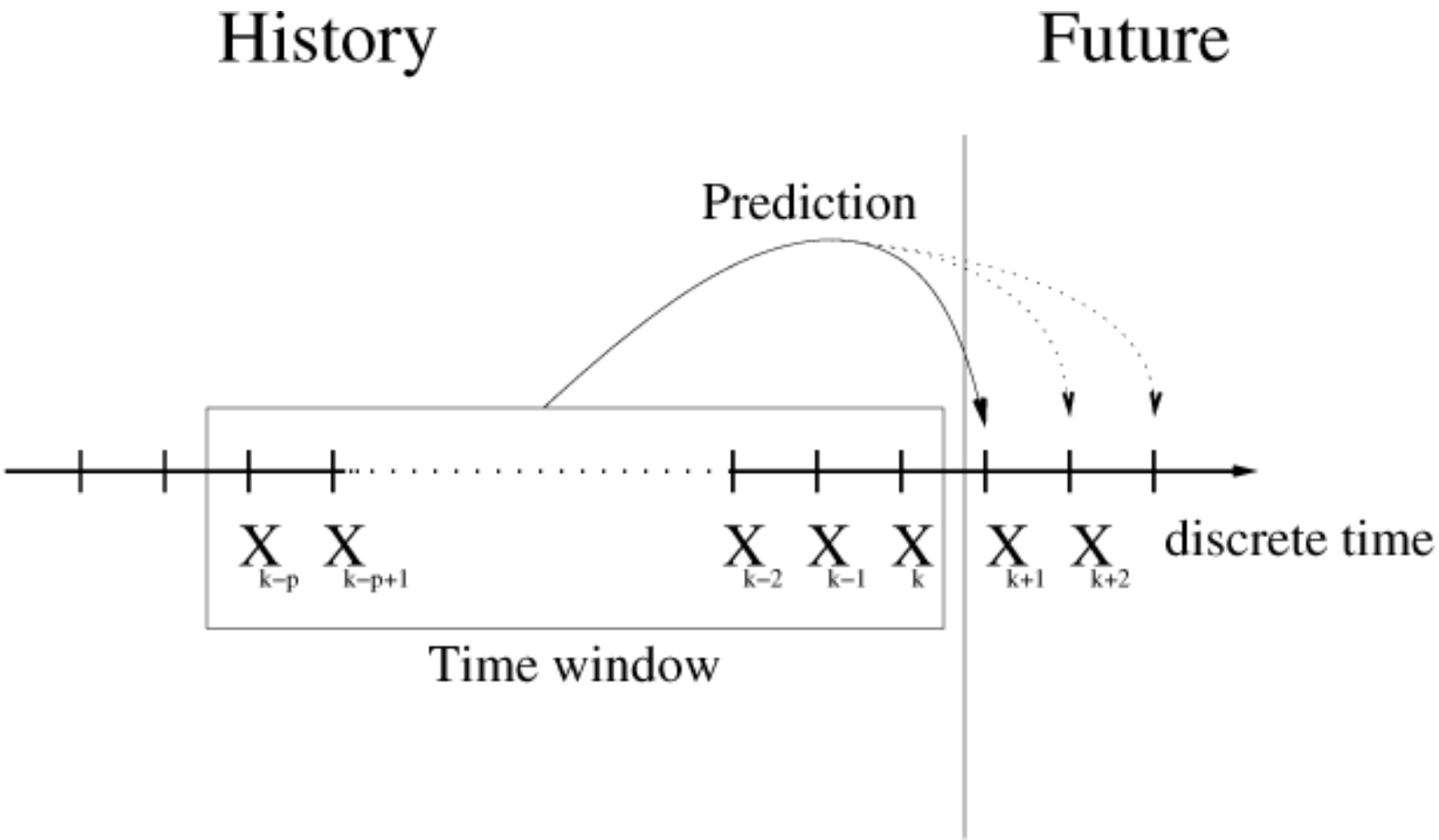
PCA clustering



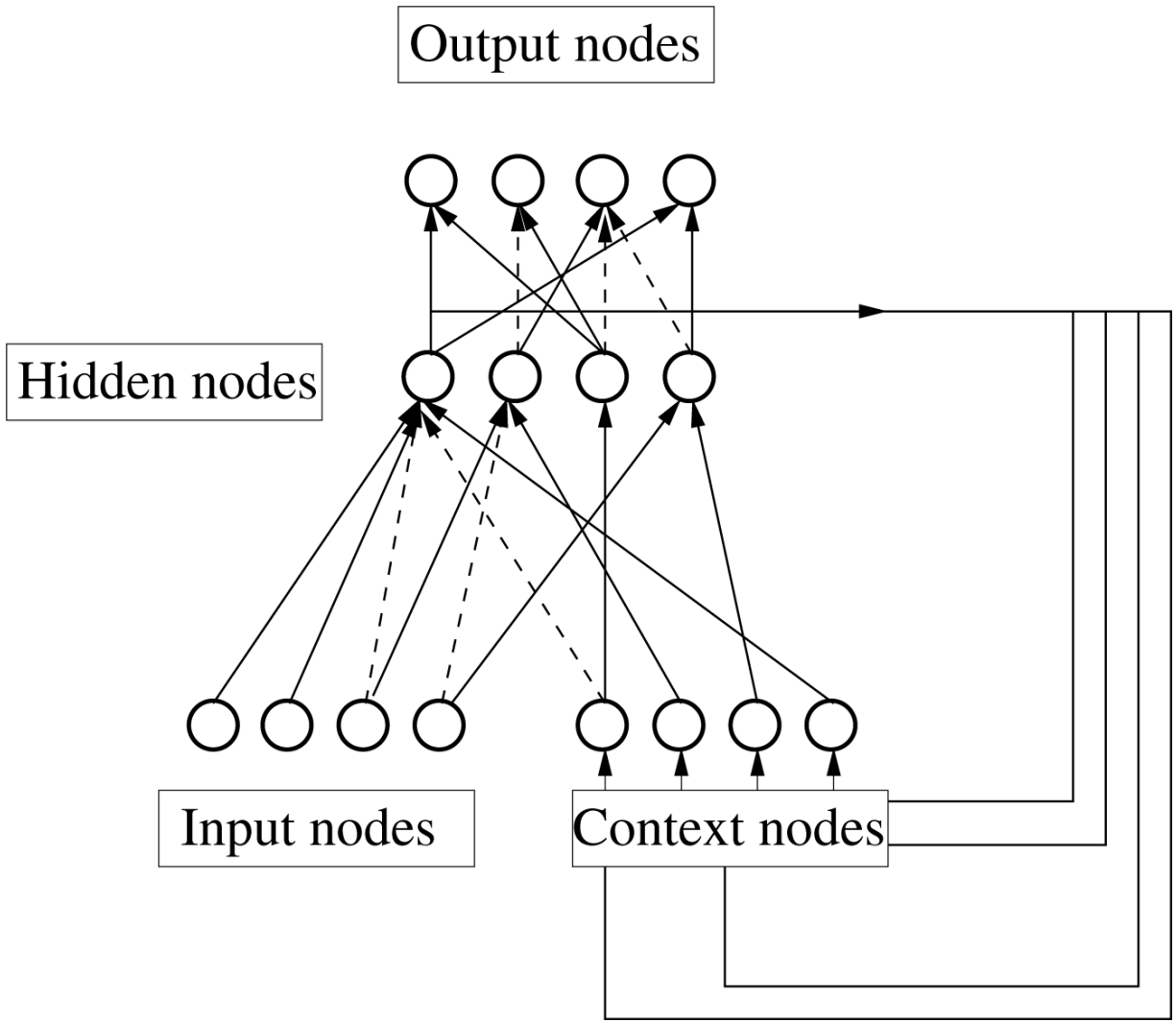
PCA path tracking



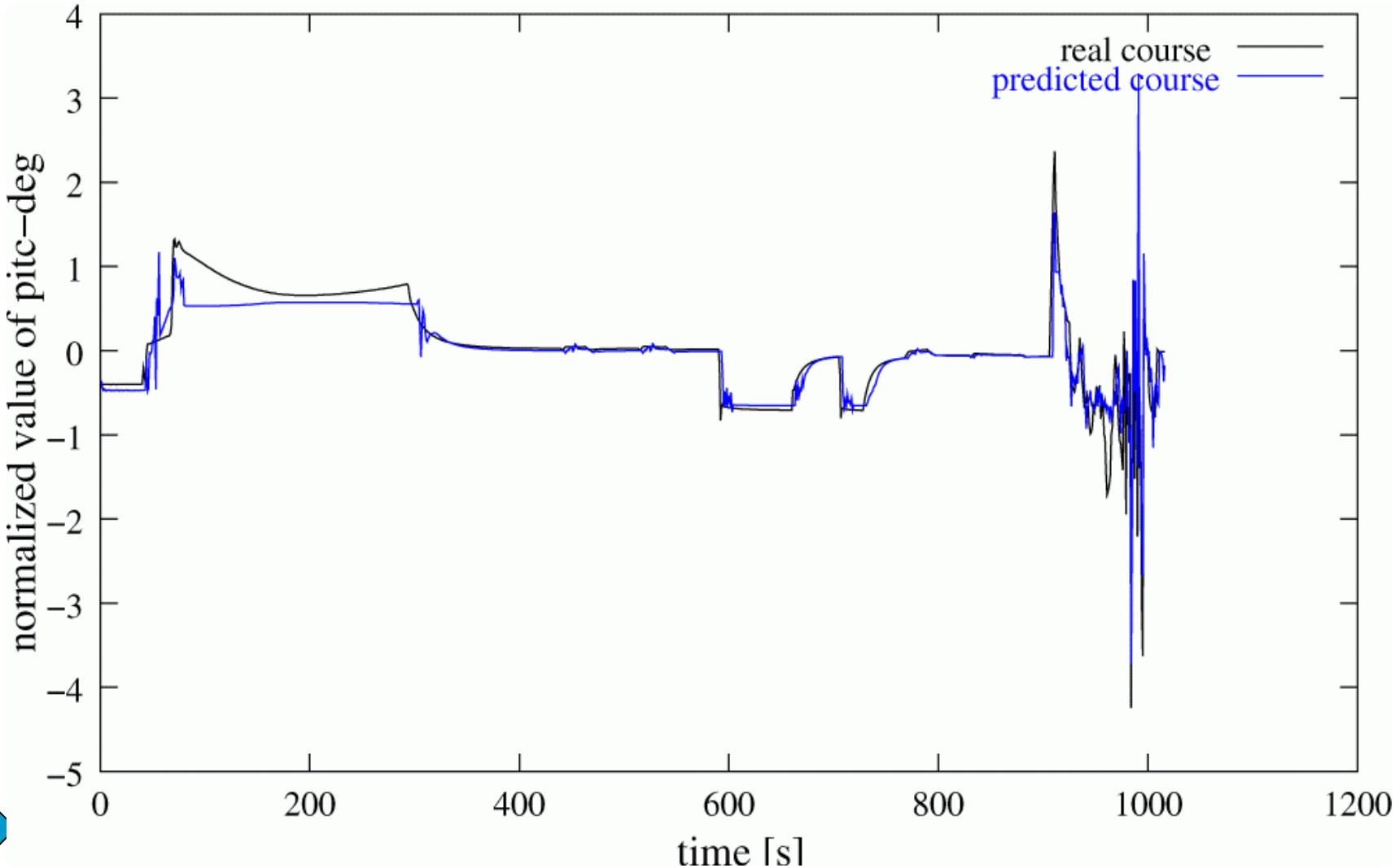
Future state prediction



Elman neural network



Elman neural network



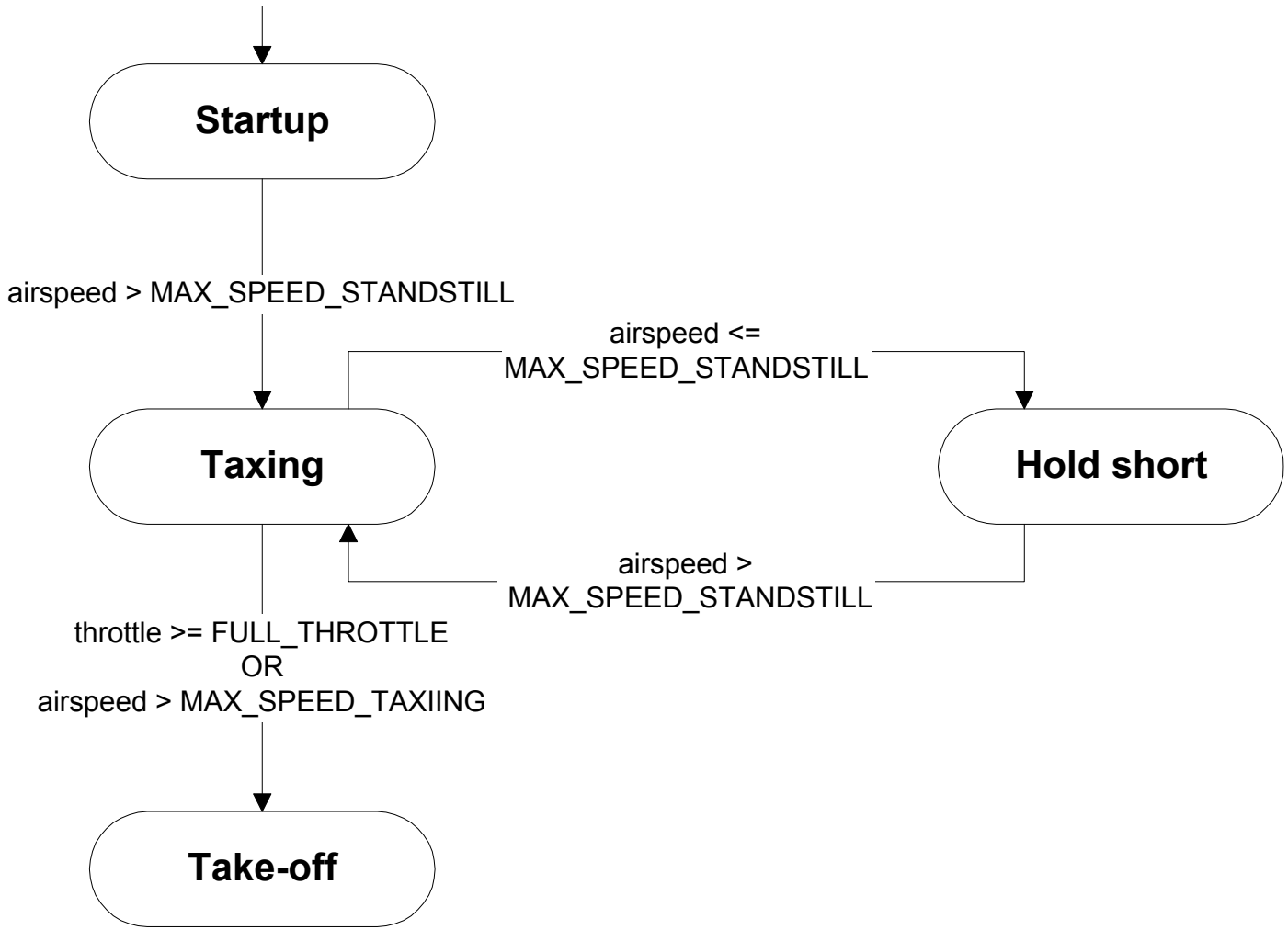
Knowledge based SR

- Real-time, on-line interpretation
- Uses state-transitions and production rules
  - Easy to adjust
  - Interpretation is transparent
  - More detailed situation recognition
  - Multiple types of airplanes (XML files)

Knowledge based SR

- Recogniser output:
  - High-level situation(s) (start, landing, etc.)
  - Expected actions (push throttle, set radar,etc.)
  - Recognised actions

High-level situation: example STD



TU Delft

Expected actions: example rule

- Situation: Dogfight

  -> Set master arm

  Check HUD

  Call on radio

  Set IFF off

Results and conclusions

- Airplane sensor data can be clustered with PCA
- Prediction of simple future states with Elman neural network possible
- Rule-based system gives excellent and flexible high-level situation recognition

Future work

- Add pilot actions recognition
- Comparison with flight plan
- Add probability values for reasoning about concurrent situations/actions