

# Reactive Scheduling of Computational Resources in Control Systems

Hodai Goldman

Ben-Gurion University of the Negev  
Department of Computer Science

January 1, 2018

## 1 Automata-based Scheduling

- Motivation
- Component-based Architecture
- Büchi Games Interface
- Sub-Summery: Component Definition

## 2 Integration with Kalman

- Guiding Concept
- Guided Tour Simulation

## 3 Experiment with real-life case-study

- The Mission
- Simplifying the Kalman filter with complementary filter
- Results

## 4 Conclusion

- Conclusion
- Related Work

# Outline

## 1 Automata-based Scheduling

- Motivation
- Component-based Architecture
- Büchi Games Interface
- Sub-Summery: Component Definition

## 2 Integration with Kalman

- Guiding Concept
- Guided Tour Simulation

## 3 Experiment with real-life case-study

- The Mission
- Simplifying the Kalman filter with complementary filter
- Results

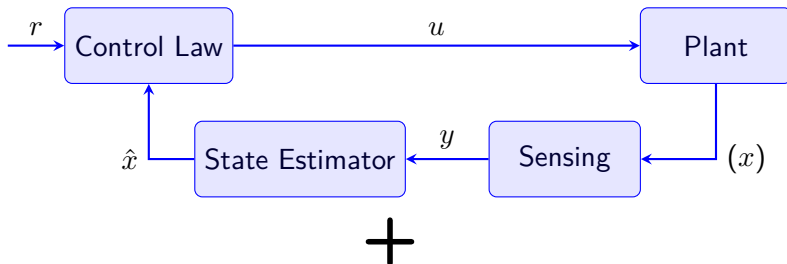
## 4 Conclusion

- Conclusion
- Related Work

# An control problem example

*present the example: robot moving in root with obstacles, mission 1: avoid obstetrical(camera), mission 2: follow the guiding root(GPS)*

# The Traditional Solution



Constant time steps + periodic tasks

*time steps*

*figure* +

Task	Period	Deadline
Check for obstacles	10ms	1.5ms
Check GPS position	10ms	0.5ms
Control Law	2ms	0ms
...		

# The Main Software Design Problems

Task	Period	Deadline
Check for obstacles	10ms	1.5ms
Check GPS position	10ms	0.5ms
Control Law	2ms	0ms
...		

## The design problems from our point of view

- **All the tasks are highly coupled:** *any change or addition of some task require to consider all other tasks requirements*
- **Static and inefficient scheduling:** *the table is defined for the worst case talk about related work on this direction*
- **No consideration of the environmental conditions:** *it is a cyber-physical system after all*

# The Goal

In this thesis we design an **reactive** scheduling framework for real-time systems

## Required features:

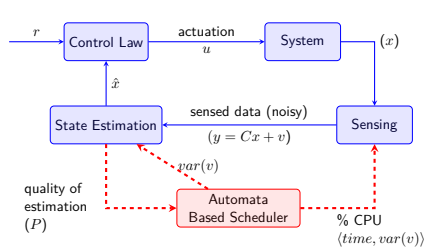
- **Independent** and **composable** requirements
- **Control objective based** requirement interface
- Environment **adoptive** scheduler

# Outline

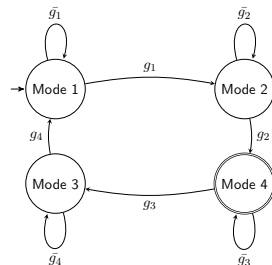
- 1 Automata-based Scheduling
  - Motivation
  - Component-based Architecture
  - Büchi Games Interface
  - Sub-Summery: Component Definition
- 2 Integration with Kalman
  - Guiding Concept
  - Guided Tour Simulation
- 3 Experiment with real-life case-study
  - The Mission
  - Simplifying the Kalman filter with complementary filter
  - Results
- 4 Conclusion
  - Conclusion
  - Related Work



# The Proposed Architecture



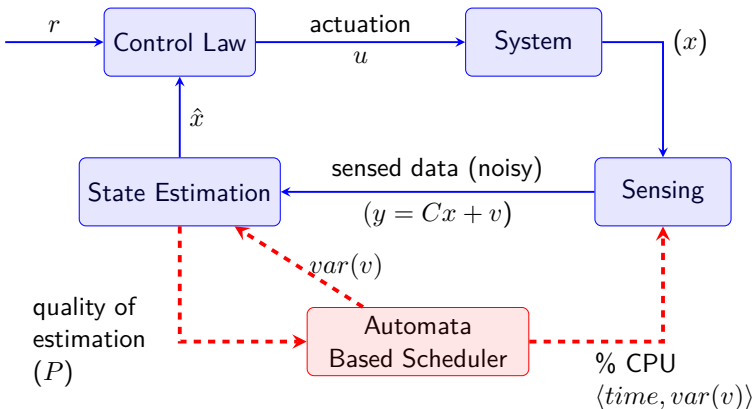
+



# System Design

## The Proposed Architecture

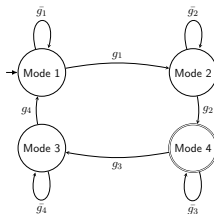
*Explain that the scheduler is involved in the control loops*



# Automata-Based Specification Interface

## The Proposed Architecture

*maybe add a word about RTcomposer and GameComposer*



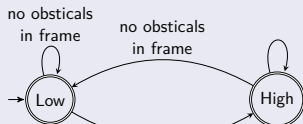
## Why Automata

- **Lite:** minimal resource consumption at run-time
- **Composable:** easy to compose independent components
- **Automata theory built in:** allows for tools such *GOAL*
- **Expressiveness**

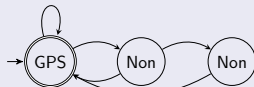
# Example of Guarded Automata

## The Proposed Architecture

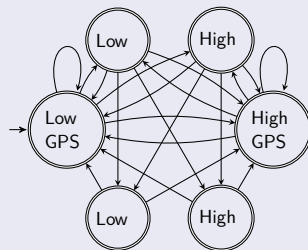
### Obstacle avoidance component



### GPS navigation component



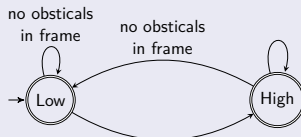
### Composed guarded automata



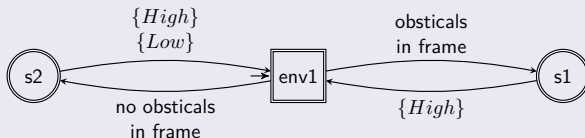
# Simplifying the Guarded Automata

## The Proposed Architecture

### Mode-based guarded automata (for good intuition)



### The automata in practice (best match $\omega$ -word theory)



**Q: How to create the guarded automata?** By wining Büchi games

# Outline

## 1 Automata-based Scheduling

- Motivation
- Component-based Architecture
- Büchi Games Interface
- Sub-Summery: Component Definition

## 2 Integration with Kalman

- Guiding Concept
- Guided Tour Simulation

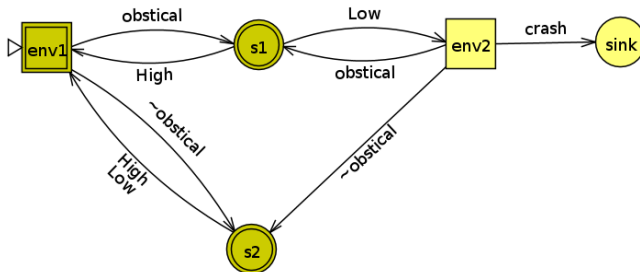
## 3 Experiment with real-life case-study

- The Mission
- Simplifying the Kalman filter with complementary filter
- Results

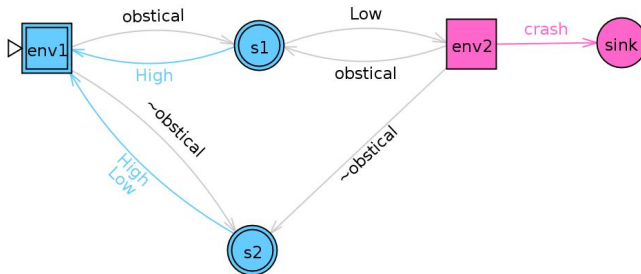
## 4 Conclusion

- Conclusion
- Related Work

# Büchi game remainder



# Büchi game remainder





# Outline

## 1 Automata-based Scheduling

- Motivation
- Component-based Architecture
- Büchi Games Interface
- Sub-Summery: Component Definition

## 2 Integration with Kalman

- Guiding Concept
- Guided Tour Simulation

## 3 Experiment with real-life case-study

- The Mission
- Simplifying the Kalman filter with complementary filter
- Results

## 4 Conclusion

- Conclusion
- Related Work



*technical*

# Outline

- 1 Automata-based Scheduling
  - Motivation
  - Component-based Architecture
  - Büchi Games Interface
  - Sub-Summery: Component Definition
- 2 Integration with Kalman
  - Guiding Concept
  - Guided Tour Simulation
- 3 Experiment with real-life case-study
  - The Mission
  - Simplifying the Kalman filter with complementary filter
  - Results
- 4 Conclusion
  - Conclusion
  - Related Work

*Explain the concept of estimate the errors*

# Outline

- 1 Automata-based Scheduling
  - Motivation
  - Component-based Architecture
  - Büchi Games Interface
  - Sub-Summery: Component Definition
- 2 Integration with Kalman
  - Guiding Concept
  - Guided Tour Simulation
- 3 Experiment with real-life case-study
  - The Mission
  - Simplifying the Kalman filter with complementary filter
  - Results
- 4 Conclusion
  - Conclusion
  - Related Work



*the simulation*

# Outline

- 1 Automata-based Scheduling
  - Motivation
  - Component-based Architecture
  - Büchi Games Interface
  - Sub-Summery: Component Definition
- 2 Integration with Kalman
  - Guiding Concept
  - Guided Tour Simulation
- 3 Experiment with real-life case-study
  - The Mission
  - Simplifying the Kalman filter with complementary filter
  - Results
- 4 Conclusion
  - Conclusion
  - Related Work

1. mission definition
2. scheduling objectives
3. how we review the results (the  $x$  axis)
4. add a video



# Outline

- 1 Automata-based Scheduling
  - Motivation
  - Component-based Architecture
  - Büchi Games Interface
  - Sub-Summery: Component Definition
- 2 Integration with Kalman
  - Guiding Concept
  - Guided Tour Simulation
- 3 Experiment with real-life case-study
  - The Mission
  - Simplifying the Kalman filter with complementary filter
  - Results
- 4 Conclusion
  - Conclusion
  - Related Work

1. *why not Kalman*
2. *how we use complementary filter*
3. *the linearize model in  $x$  / roll axis*
4. *update state (equations)*

# Outline

- 1 Automata-based Scheduling
  - Motivation
  - Component-based Architecture
  - Büchi Games Interface
  - Sub-Summery: Component Definition
- 2 Integration with Kalman
  - Guiding Concept
  - Guided Tour Simulation
- 3 **Experiment with real-life case-study**
  - The Mission
  - Simplifying the Kalman filter with complementary filter
  - **Results**
- 4 Conclusion
  - Conclusion
  - Related Work



*the automata and their results*

# Outline

- 1 Automata-based Scheduling
  - Motivation
  - Component-based Architecture
  - Büchi Games Interface
  - Sub-Summery: Component Definition
- 2 Integration with Kalman
  - Guiding Concept
  - Guided Tour Simulation
- 3 Experiment with real-life case-study
  - The Mission
  - Simplifying the Kalman filter with complementary filter
  - Results
- 4 Conclusion
  - Conclusion
  - Related Work

*instead of with Related Work*

# Outline

- 1 Automata-based Scheduling
  - Motivation
  - Component-based Architecture
  - Büchi Games Interface
  - Sub-Summery: Component Definition
- 2 Integration with Kalman
  - Guiding Concept
  - Guided Tour Simulation
- 3 Experiment with real-life case-study
  - The Mission
  - Simplifying the Kalman filter with complementary filter
  - Results
- 4 **Conclusion**
  - Conclusion
  - **Related Work**

*review of similar papers: A table with few papers*



# Thanks