Reactive Scheduling of Computational Resources in Control Systems

Gera Weiss
Ben-Gurion University
Beer-Sheva, Israel
geraw@cs.bgu.ac.il

Hodai Goldman
Ben-Gurion University
Beer-Sheva, Israel
hodaig@cs.bgu.ac.il

ABSTRACT

1. INTRODUCTION

Cyber-physical systems, where computers interact with and control physical entities ar at the heart of many critical applications. In a typical cyber-physical application, such as robot control, there are many control loops, responsible for various aspects of the system, that run simultaneously and share the same computational resources. Thanks to the ever growing computational power, engineer often tend to ignore this shared resource and design the controllers as if there is

2. ARCHITECTURE: AUTOMATA BASED SCHEDULER

Sections go here.

3. SIMULATIONS

4. PROPOSED METHODOLOGY

1. **Controller design:** Based on the separation principle, we propose to design the controller to achieve the control objectives assuming a perfect observation. In practice, this may not be feasible because controller designs such as PID require a system to experiment with. In this case, as demonstrated in the case-study below (see Section 5), we propose to work with one of the observation modes. If the system is close to linear, this should result with a near optimal design.

2. **Observers design:** Specification of sensor modes and observers design

3. **Performance analysis:** Now, we can perform some experiments with the different observer modes and analyze transient behaviors. Specifically, as shown in the case study **??**, we can measure how long it takes for the error to accumulate after switching to a lesser observation mode and formulate how this error affects the control objectives.

4. **Scheduling automata design:** Base on the analysis we can specify the resource scheduling requirements in the form of *specification automaton*. The goal is to design flexible specification that allow dynamic scheduling in order to adapt the environment and the system state, this will improve the system efficient.

5. APPLICATION: STABILIZING A QUADROTOR IN FRONT OF A WINDOW

The case study we used to test our concept is the development of a controller that stabilizes a quadrotor [**?**] in front of a window. We implement an autonomous controller for that task and evaluated its performance.

The part of the controller that we focused on is the vision based position controller. Specifically, the main controller, that we will describe below, uses a standard low-level angular controller and a simple image processing algorithm that identifies the position of the corners of the window in the image plane[1]. Its goal is to regulate the position of the quadrotor by tilting it. Note that rotations of the quadrotor generate a more-or-less proportional accelerations in a corresponding direction. A main challenge for this controller is that the computer vision algorithm takes significant time to compute relative to the fast control loop. We can decrease computation time by lowering the resolution, but this also increase the measurement noise. We will demonstrate how

[1]In the experiment, to simplify the image processing algorithm, we marked the corners of the window with led lights.

adaptive scheduling of the resolution can serve for balancing resource consumption vs. control performance.

5.1 Observer Design

We first implemented an observer that gets the positions of the widow corners and use them to estimate the position of quadrotor relative to the window.

5.2 Controller Design

We used a simple Proportional Integral Derivative (PID) controller that we tuned by experimenting with the quadrotor in our lab. Based on the separation principle [?], we tuned the parameters of our controller only with the highest resolution observer and used them for all the other observers without any modification.

5.3 Controller Design

The drone is controlled by a Raspberry pi with the navio [?, ?] that runs a modified ArduPilot [?] software. We implement standard PID [1] controller that regulates the position and attitude of the drone relative to the window. We used inertial sensors (Gyroscope and Accelerometer) for attitude control relative to earth, and we use camera [?] and simple image processing for position and attitude relatively to the window **??**. The image processing have few different operation modes which have different image resolution, better resolution image produce more accurate measurement but consume more processing time, changing operation modes allows us to control the trade off between mesurment quality and processing time as shown in Table **??**, for simplicity we examine only two modes, *High quality* with resolution of 960p and *Low quality* with resolution of 240p.

As shown in the simulation at Section 3 we should use Kalman filter estimator, in our case we have non-linear system... For simplicity, inspiring from Kalman filter [?], we use two steps estimator that *predict* the current state evolved from the previous state using the system model ($x_{k|\hat{k-1}}$) and then *update* the prediction with current state measurement from image (z_k). The result estimation ($x_{\hat{k|k}}$) is weighted average: $x_{\hat{k|k}} = Kx_{k|\hat{k-1}} + (1-K)z_k$, the K gain is defined different for each resolution for achieving estimation in each resolution.

5.4 Analysis and Specification Automata

The objective of the system is to maintain stable hovering in front of the window. Hence, the performances of the system is measured by the amount of deviation from the center line of the window in the critical axis x (see Figure **??**). Our goal is to achieve maximum performance with minimal amount of processing (CPU percentage). The main consumer of computation resources in this system is the heavy observation tasks.

Analyzing the test results shown in Table **??** we see that High quality observation mode provide 0.4 meter error tolerance meaning we can fly this mode in 1.2 meter wide corridor[2], with the cost of 30% CPU usage. With adaptive scheduling specification we lower the CPU usage without significant worsening of High quality performances.

From the Low quality experiment graph we can see that measurement post-fit residual ($\tilde{y}_{k|k}$) is accumulate proportional to the deviation from the center of x axis, and we can use $\tilde{y}_{k|k}$ vales to predict

TODO - position ($x_{k|k}$) derived schedulers

TODO - complex / agregated schedulers

TODO - gyro derived??

5.5 Results

TODO - graphs and tables

6. EXPERIMENT SETUP

7. CONCLUSIONS

Conclusion goes here.

Acknowledgments

Acknowledgement goes here.

8. REFERENCES

[1] K. J. Åström and T. Hägglund. *Advanced PID control*. ISA-The Instrumentation, Systems and Automation Society, 2006.

APPENDIX

Appendix goes here.

[2] Experiments was done with 0.4 meter wide quad-rotor