Memory-Aware Feedback Scheduling of Control Tasks

Sven Gestegård Robertz
Dan Henriksson
Anton Cervin

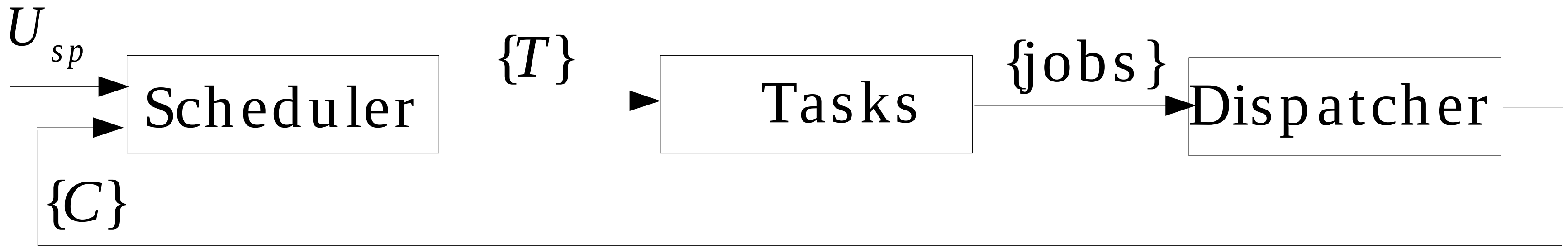
Lund University
Sweden

Outline

- Feedback scheduling
- Automatic memory management / GC scheduling
- GC-aware period assignment
- Controlling the allocation rate
- Conclusions

Feedback scheduling

- Dynamic resource allocation
- Worst-case design too pessimistic
- Use feedback to handle uncertainties
- Deadline misses can be viewed as disturbances
- Control to keep utilization below a set-point



Feedback Scheduling

- Optimization problem

$$min \sum_{i=1}^{n} J_i(f_i)$$
$$subject\ to \sum C_i f_i \le U_{sp}$$

- Closed-form solution for linear cost functions

$$f_i^* = (\frac{\gamma_i}{C_i})^{\frac{1}{2}} \frac{U_{sp}}{\sum_{j=1}^{n} (C_j \gamma_j)^{(\frac{1}{2})}}$$

- Similar for quadratic cost functions

Automatic Memory Management

- Safe Object-Oriented Languages
- Real-time GC scheduling
- Time-triggered GC
  - Non-intrusive GC
  - Concurrent GC scheduled as a separate task
  - GC scheduled by the standard task scheduler
  - Explicit deadline for each GC cycle

Time-Triggered GC

- Deadline determined
  - a priori

$$T_{GC} \leq \frac{\frac{H - L_{max}}{2} - \sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} f_i a_i}$$

  - adaptive at run-time

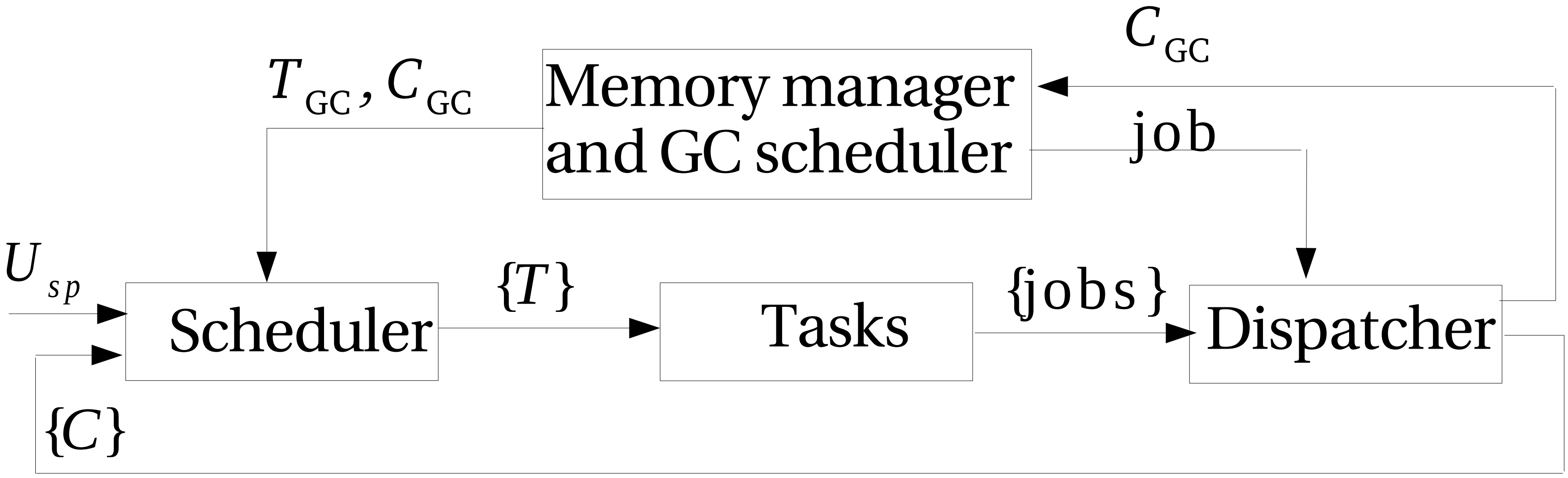
- GC cycle time depends on allocation rate

Problem Statement

- Managing CPU time and memory
  - Global resource management problems
  - Cannot be treated independently
    - Memory operations use CPU time
    - Automatic memory management
      - No explicit memory management cost
      - Feedback scheduling requires total CPU cost / task
  - Changing task periods affects memory usage

GC-Aware Period Assignment

- GC jobs dispatched as any other jobs
- The scheduler cannot change the GC period



GC-Aware Period Assignment

- Two models
  - Separate: use GC scheduler as reference generator

$$U_{ref} = U_{sp} - U_{GC}$$

$$U_{GC} = \frac{C_{GC}}{T_{GC}}$$

  - Integrated: incorporate GC work into FBS problem

$$\sum \frac{C_i + K_{GC} a_i}{h_i} \leq U_{sp}$$

- Biggest difference: fairness

Controlling the Allocation Rate

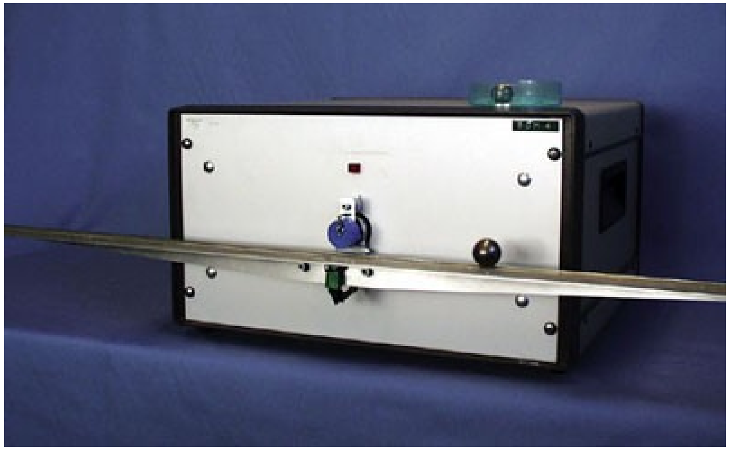
- Priorities for memory allocations
  - Tasks have mandatory and optional parts
  - If low-on-memory, turn off optional parts
- Allows (some) control of the allocation rate
- Extend the optimization problem

$$\min \sum_{i=1}^{n} J_i(f_i, \mathbf{a}^{opt})$$

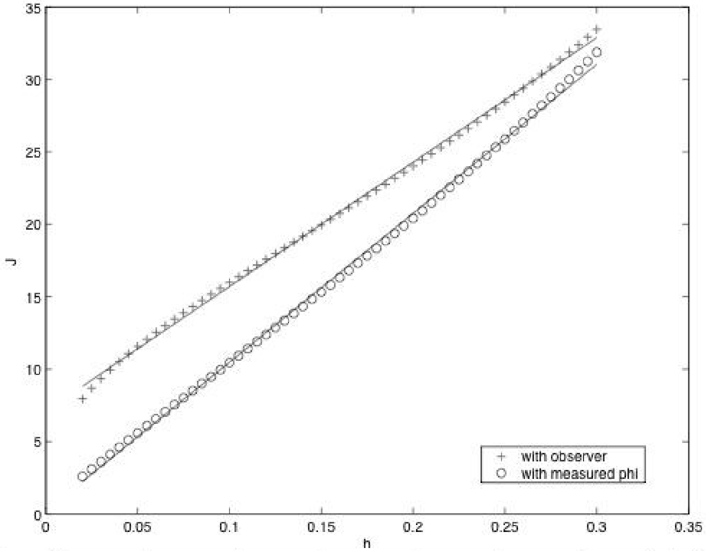
$$\text{subject to} \sum \left(C_i + \mathbf{K}_{GC} (\mathbf{a}_i^m + \mathbf{a}_i^{opt}) \right) f_i \le U_{sp}$$

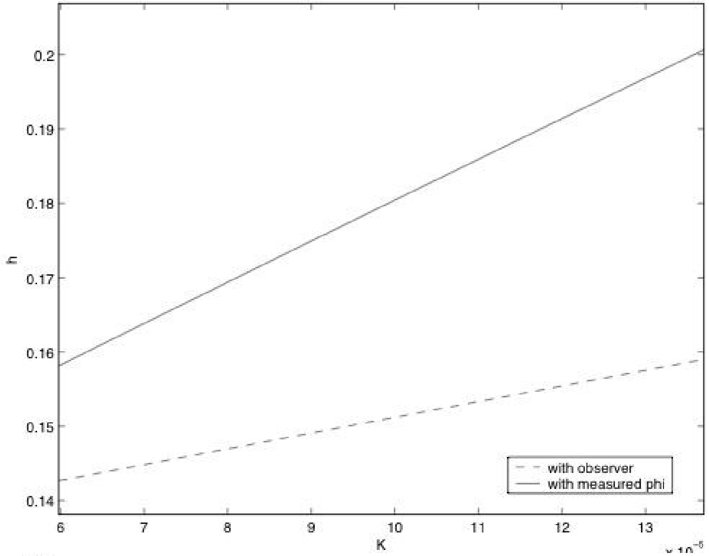
Case study

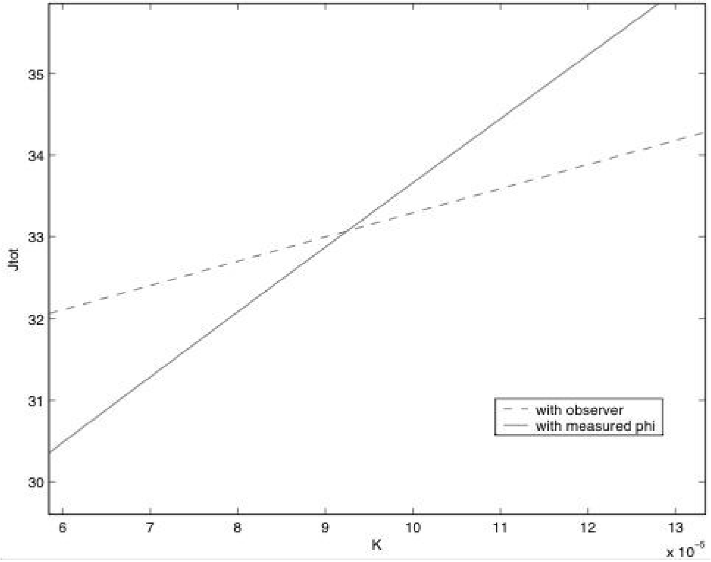
- One control task + one disturbance task
- Ball-and-beam process
  - Position measurement
  - Angle measurement opt.
- Cost analysis
  - With and without angle measurement
  - For different GC costs (= sampling rates)
- Simulation

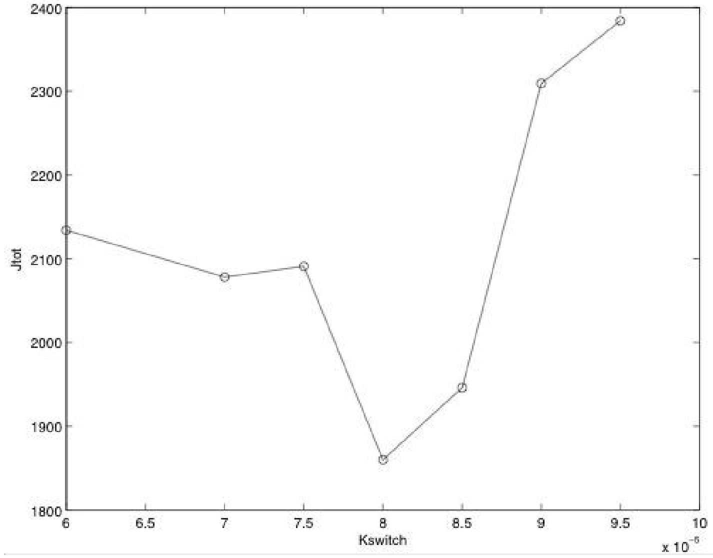


Results


35
30
25
20
15
10
5
0
J
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
h
+ with observer
○ with measured phi


0.2
0.19
0.18
0.17
0.16
0.15
0.14
h
6
7
8
9
10
11
12
13
K
x 10^-5
with observer
with measured phi


35
34
33
32
31
30
Jtot
6
7
8
9
10
11
12
13
K
x 10^-5
with observer
with measured phi


2400
2300
2200
2100
2000
1900
1800
Jtot
6
6.5
7
7.5
8
8.5
9
9.5
10
Kswitch
x 10^-5

Conclusions

- CPU time and memory "co-managed"
- Necessary for FBS in system with GC
- Take memory management overhead into account in period assignment