A Dual-Mode Strategy for Performance-Maximisation and Resource-Efficient CPS Design

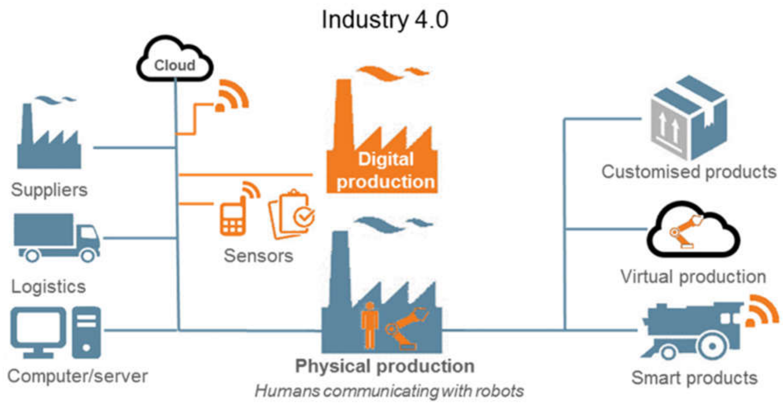
Xiaotian Dai, Wanli Chang, Shuai Zhao, Alan Burns

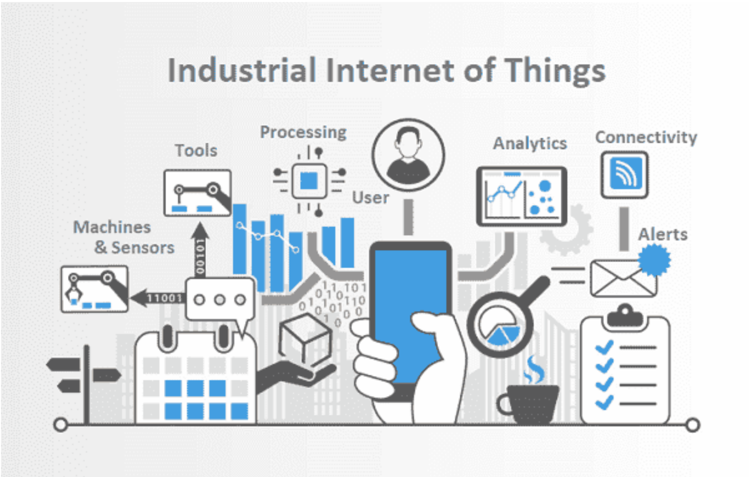
Xiaotian Dai

Real-Time Systems Research Group (RTSRG)
University of York
wanli.chang@york.ac.uk

Introduction

- Definition of Cyber-Physical Systems (CPS)
  - CPS = Control + Computing + Communication





Introduction

- Many new challenges for engineering of CPS:
  - System level design methodology and implementation;
  - Safety, reliability and security;
  - Communication and integration;
  - Monitoring and maintenance;
- These will increase the complexity of the software, and consequently demand more computational resources.

Introduction

- In cyber-physical systems, efficiently use of the resources, while guarantee system integrity is an important requirement.
- For a CPS, challenges in task and resource scheduling include:
  - Due to cost and size limitation, computational resource is often constrained;
  - Timing predictability under internal and external uncertainty;
  - Performance maximization v.s. implementation constraints;
  - Resource efficiency under high computation demand but pessimistic hypothesis.

Introduction

- Current practice for scheduling control tasks:
  - Release the control task periodically with a constant period;
  - This period is defined by a control engineer when the controller is designed. The system integration engineer then decides how to allocate and schedule the task.
- This separation of design is easy for implementation and analysis, however:
  - Control and scheduling designs are isolated;
  - The controller is unaware of underlying scheduling resources;
  - The scheduler cannot utilize the information of control performance.

Introduction

- Requirements of task scheduling in a high-efficiency CPS:
  - Control and scheduling should be cooperatively designed;
  - The design of the controller should consider the resource limitation of the processor;
  - The schedule of tasks should realize the effect of task timing to the control performance;
  - In some circumstance, trade-off between control and schedulability has to be made.

Related Work

- **Elastic task scheduling**
  - + Task period is flexible, i.e., defined as an interval;
  - + Task period is adjustable at run-time;
  - - Control parameters need to be calculated after every period change.
- **(*m*, *k*)-firm task model**
  - + Guaranteed to execute *m* in every *k* jobs;
  - - Hard to analysis control performance.
- **Dynamic period assignment**
  - + Adjust control period based on current system states;
  - + Optimal in terms of maximized summed control performance;
  - - High dynamism and relative large run-time overhead.

Our Method - DUAL

- In this work, we propose a scheduling strategy in the context of control scheduling co-design in CPS.
- This strategy, *DUAL*, uses a switching task model with two period modes.
- The two periods are assigned by the scheduler during different control phases.
- The switching time instant is pre-calculated so it is deterministic, which gives guaranteed maximal resource usage.
- Depending on the design objective, this strategy can be optimized either for either for resource saving or control performance.

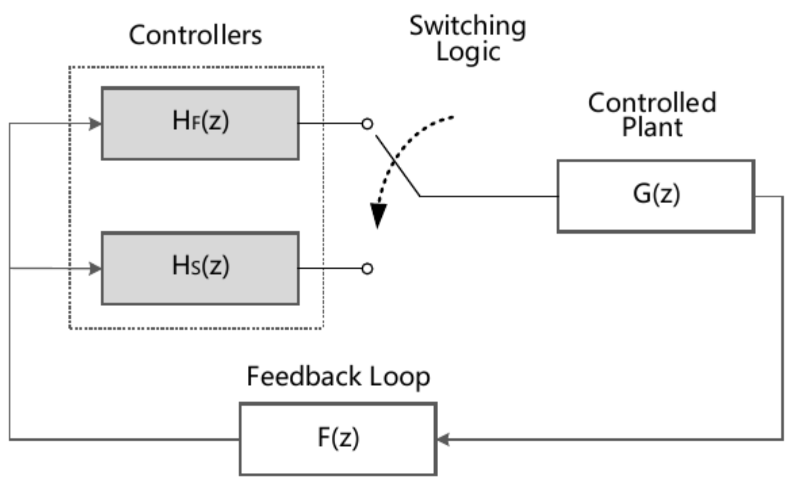
Scheduling Model

- Scheduling policy: fixed-priority scheduling
- Task priority is assigned with deadline monotonic

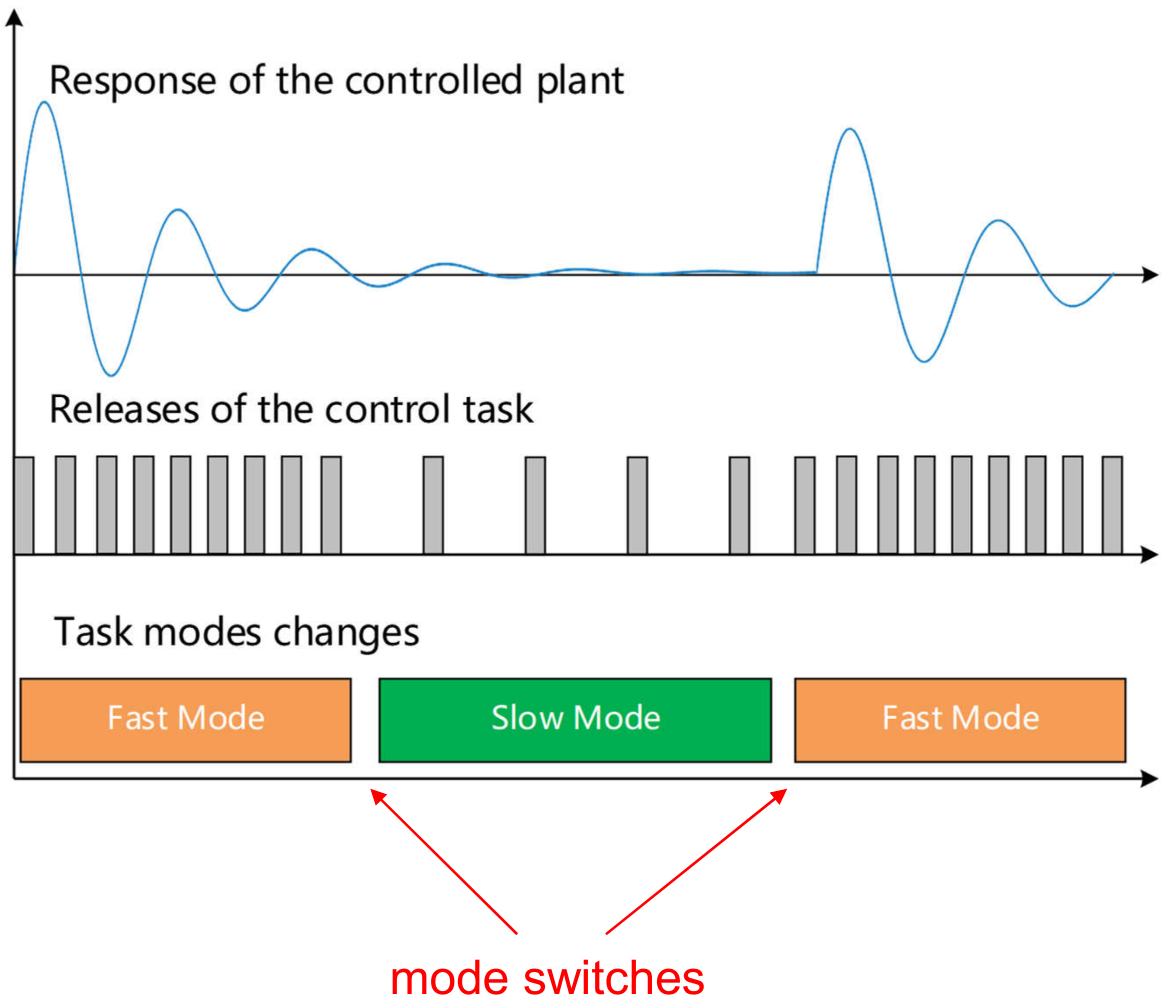
- A control task is modelled as:

$$\tau_i \equiv (\mathcal{M}_i, C_i, T_i^H, T_i^L, \alpha_i, D_i, T_i^\Gamma)$$

- Each control task has two execution modes: a fast-mode and a slow-mode



Execution Mode Switching



Schedulability Analysis

- The schedulability of DUAL can be checked through a tailored response time analysis (RTA).
- The RTA checks the critical instance when all tasks are released at the same time. A dual task firstly executes at T_i^H, switches to T_i^L and immediately switch back to T_i^H after the minimal switch interval.
- For FPS, tasks that have higher priorities than a dual task will not be affected and thus will have the same response time.
- If a task has lower priority than a dual task, its response time becomes:

$$R_i = C_i + nI_\Gamma + \left(\left\lceil \frac{\min(t_S, R_i - nI_\Gamma)}{T_j^H} \right\rceil + \left\lceil \frac{\max(0, R_i - nI_\Gamma - t_S)}{T_j^L} \right\rceil \right) C_j$$

Motivational Example - 1

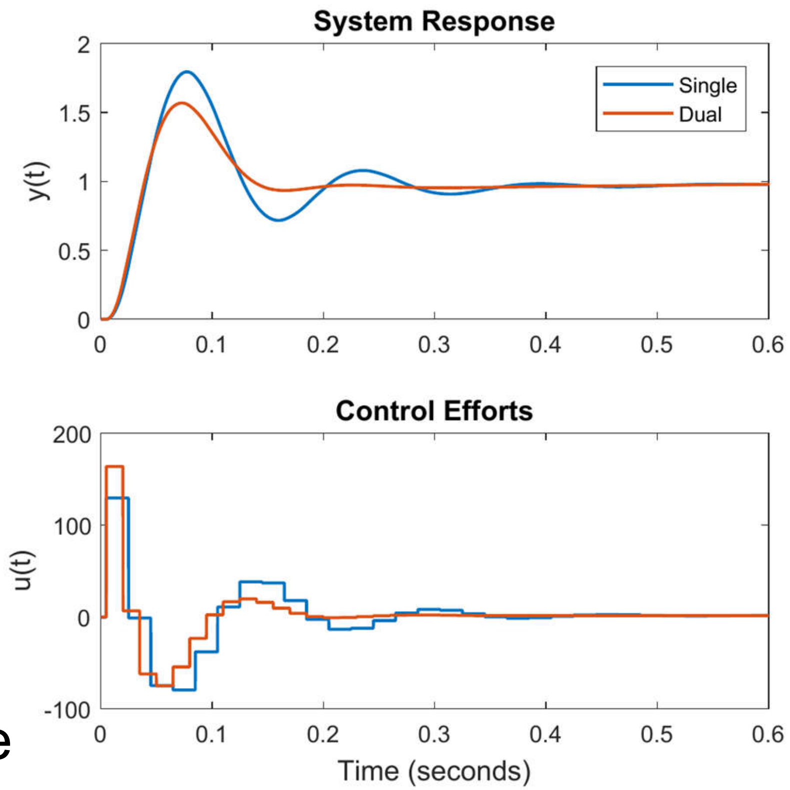
- Given a control plant:

$$G(s) = \frac{15}{s^2 - 0.2s + 25.01}$$

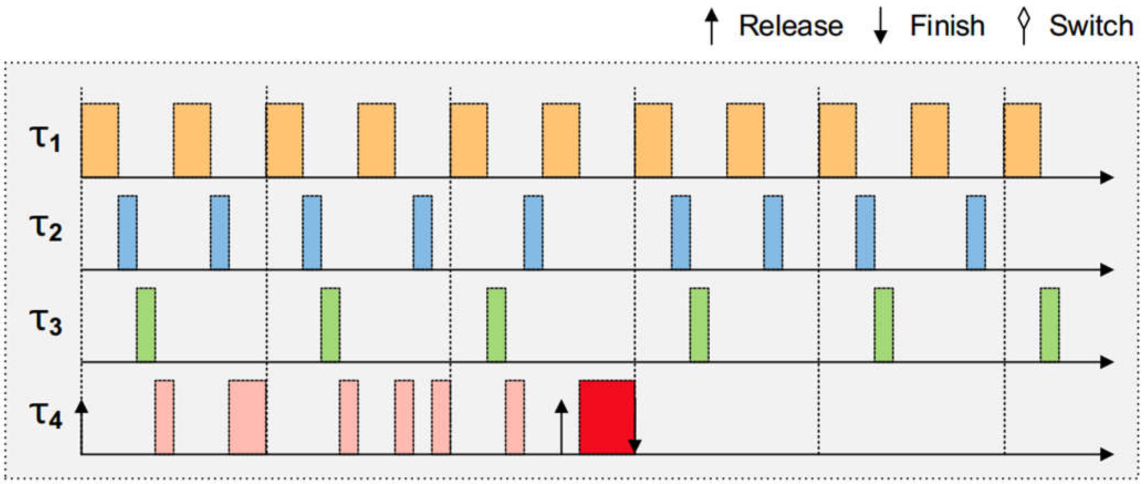
- The control performance requires:

$$T_{ss} < 0.35$$

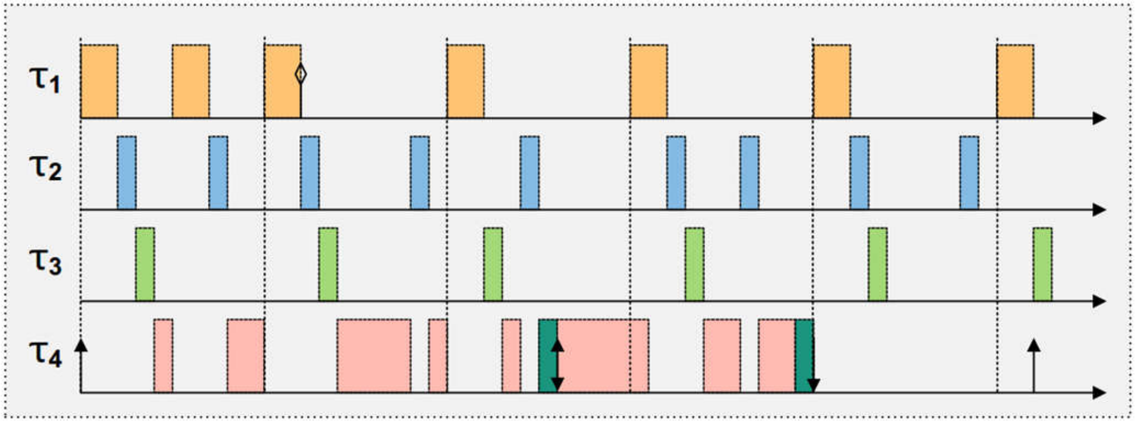
- For a control task with $T_i = 20ms$, the controller failed to satisfy the requirement.
- When using the Dual model, with $T_i^H = 15ms, T_i^L = 35ms$, and switches at $t = 30ms$. The performance is satisfied with the same utilization.



Motivational Example - 2



Task	p_i	C_i (ms)	T_i (ms)	D_i (ms)
τ_1	0	4	10	10
τ_2	1	2	12	12
τ_3	2	2	14	14
τ_4	3	20	50	50



- When making task 4 into a Dual model. The unscheduable taskset now becomes scheduable.

Task Parameters Optimization

- Each task has three configurable parameters: (T_i^H, T_i^L, α_i)
- For *N* control tasks, there will have *N* x 3 parameters to solve.
- Also these parameters are not independent and are **highly coupled**.

- The task parameters need to be determined, however the **searching dimension** of the parameter space **is high**.

Task Parameters Optimization

- To solve this problem, we formulate it into an optimization by using the genetic algorithm (GA). We formulate the following two fitness functions:

$$\lambda_C = \frac{1}{n}\sum_{i=1}^{n} P_c(i)$$

control fitness function

$$\lambda_U = 1 - \sum_{\tau_i \in \Gamma} U_i$$

utilization fitness function

s.t.:

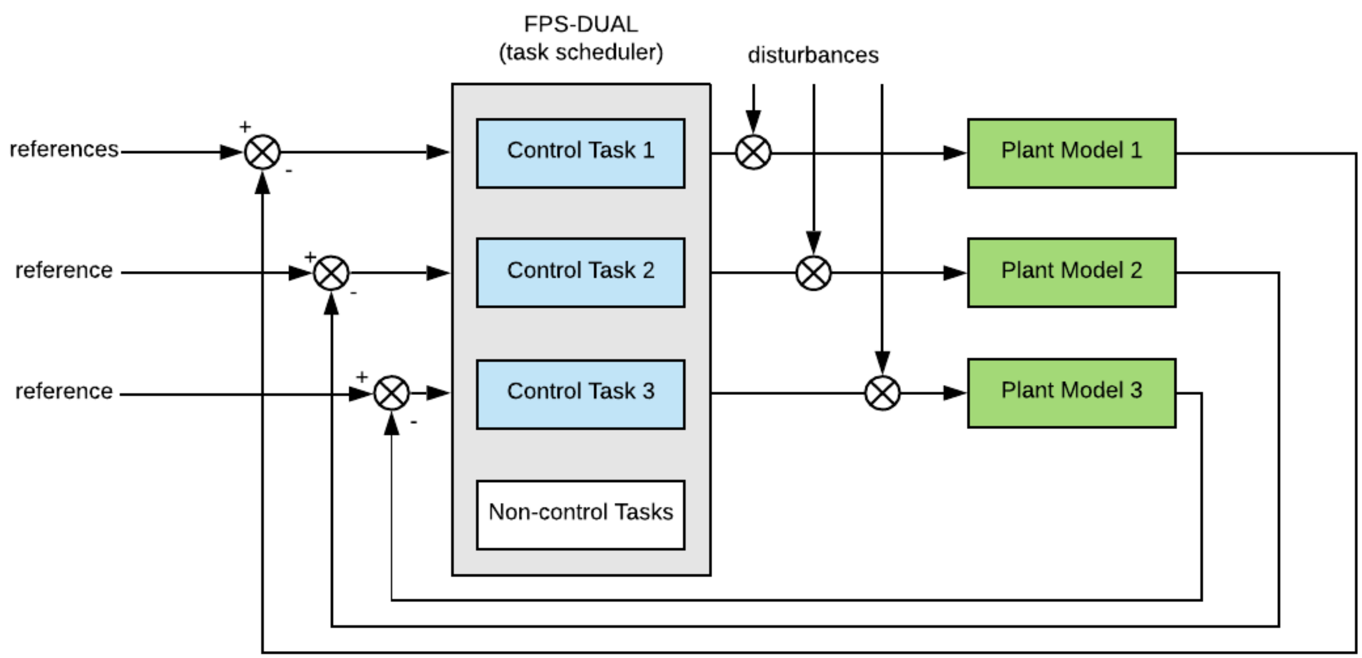
$$\begin{aligned}
&\forall i \in \Gamma_c : \\
&0 < \alpha_i \le 1 \\
&T_i^- \le T_i^H < T_i^L \le T_i^+ \\
&P_c(i) \ge 0 \\
&\forall \tau_j \in \Gamma : \\
&R_j \le D_j
\end{aligned}$$

$$\begin{aligned}
&\forall i \in \Gamma_c : \\
&0 < \alpha_i \le 1 \\
&T_i^- \le T_i^H < T_i^L \le T_i^+ \\
&0 < U_i \le 1 \\
&\forall \tau_j \in \Gamma : \\
&R_j \le D_j
\end{aligned}$$

Evaluation

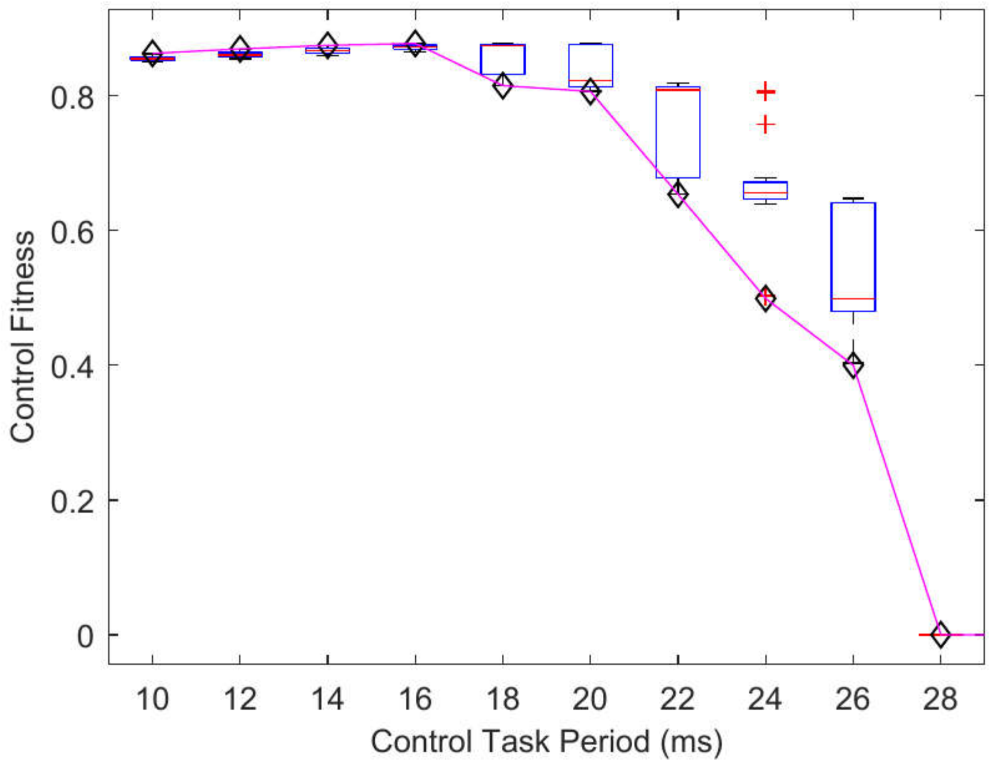
The method is evaluated using a hybrid simulation:

- Simulink for simulating continuous control system dynamics.
- C++ *S*-function for simulating discrete-time scheduling behavior.



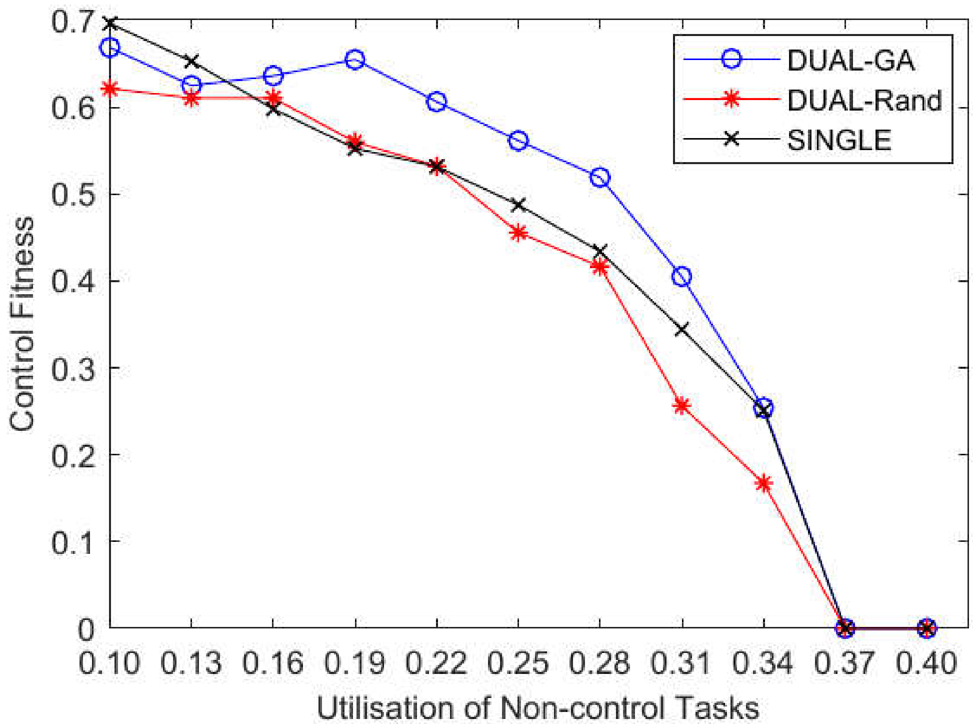
Experiment – I

Compare Dual (boxplot) with Single (diamond)



- Fitness: higher is better.
- Dual has better fitness than the single for most of the control intervals.

Experiment – II
Optimizing control performance



- Fitness: higher is better
- GA is effective compared with random searching.
- DUAL has better fitness than the single for most of the time.

Experiment – III
Optimizing resource efficiency

Table 2: Utilisation fitness λ_U of single- and dual-mode tasks with varied U_{nc}

U_{nc}	**0.10**	**0.13**	**0.16**	**0.19**	**0.22**	**0.25**
Single	0.129	0.099	0.069	0.039	0	0
Dual	0.229	0.204	0.204	0.151	0.117	0
Δ	0.100	0.105	0.136	0.112	0.117	0

- Fitness: higher is better
- $U_{nc} = 0$ means the taskset is not schedulable

Table 3: Number of additional schedulable tasks of all non-control task candidates

	Single			Dual		
U_{nc}	min	mean	max	min	mean	max
0.10	6	8.4	11	9	10.9	13
0.13	6	7.3	10	8	10.2	13
0.16	5	6.3	9	8	10.2	13
0.19	3	4.6	7	7	8.9	11
0.22	-	-	-	6	7.8	10
0.25	-	-	-	-	-	-

- # of task: higher is better

Conclusions

In this work, we have:

- Introduced a switching task model with two periods.
- Proposed an optimization framework using GA to search the optimal task parameters.
- Evaluated the strategy and the optimization framework using multiple experiments.
- Demonstrated the Dual strategy and the parameter optimization are effective.

Future Work

This work can be further explored:

- Adapt the method to Earliest Deadline First (EDF);
- Analysis the consequence of period change to other sub-systems;
- Support multiple (more than two) control modes;
- Support multiple optimization objectives simultaneously, which include cost, energy, control effort, etc.

Thank You

for Your Attention!