Jitter Propagation in Task Chains

Shumo Wang, **Enrico Bini**, Qingxu Deng, Martina Maggio

Northeastern University, Shenyang, China (China)
University of Turin (Italy)
Saarland University (Germany), Lund University (Sweden)

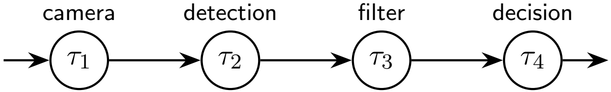
46th IEEE Real-Time Systems Symposium
Boston, MA, USA

Outline

1. Background
2. Model
3. Composition
4. Evaluation
5. Conclusion

Motivation

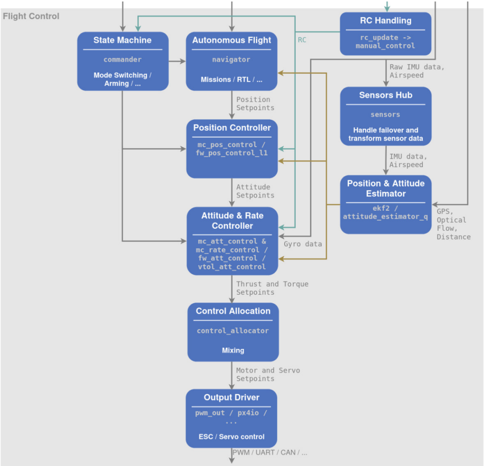
- **Use cases**
  - automotive, avionics, robotics, and UAVs
- **Functionalities delivered by concatenating tasks**
  - Camera records video
  - Object detection
  - Data filtering
  - Decision taking
- **Asynchronous communication**
  - readers do not wait for writers



camera detection filter decision

τ_1 τ_2 τ_3 τ_4

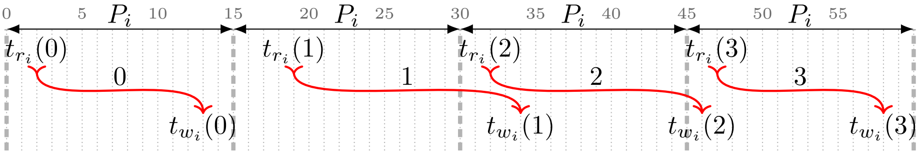
PIX4[1] UAV autopilot, flight controller



Flight Control
State Machine
commander
Mode Switching / Arming / ...
Autonomous Flight
navigator
Missions / RTL / ...
RC
RC Handling
rc_update -> manual_control
Raw IMU data, Airspeed
Position Setpoints
Sensors Hub
sensors
Handle failover and transform sensor data
Position Controller
mc_pos_control / fw_pos_control_l1
IMU data, Airspeed
Attitude Setpoints
Position & Attitude Estimator
ekf2 / attitude_estimator_q
GPS, Optical Flow, Distance
Attitude & Rate Controller
mc_att_control & mc_rate_control / fw_att_control / vtol_att_control
Gyro data
Thrust and Torque Setpoints
Control Allocation
control_allocator
Mixing
Motor and Servo Setpoints
Output Driver
pwm_out / px4io / ...
ESC / Servo control
PWM / UART / CAN / ...

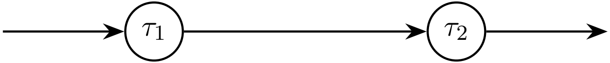
[1] source: `https://docs.px4.io`

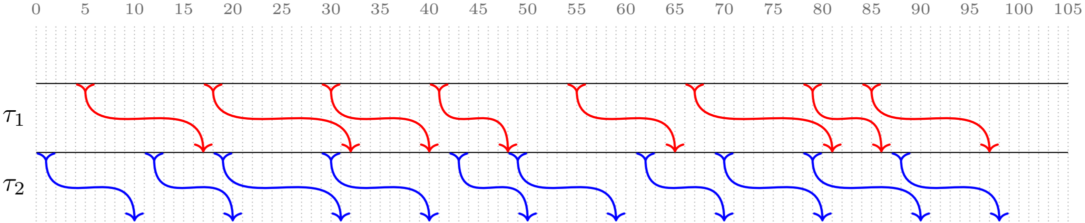
Standard model of a task τ_i in chains



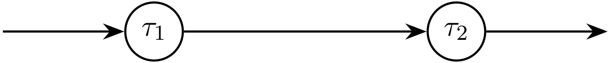
- A *task* τ_i is composed by *jobs* (curvy arrows) indexed by $0, 1, 2, \ldots$
- Jobs are released every *period* P_i
- $t_{r_i}(j)$, sequence of read instants of τ_i jobs
- $t_{w_i}(j)$, sequence of write instants of τ_i jobs
- No need of execution times, scheduling algorithm: we only care of the read/write instants
- A "task" could also be a network message, a bus line in transportation network, etc.

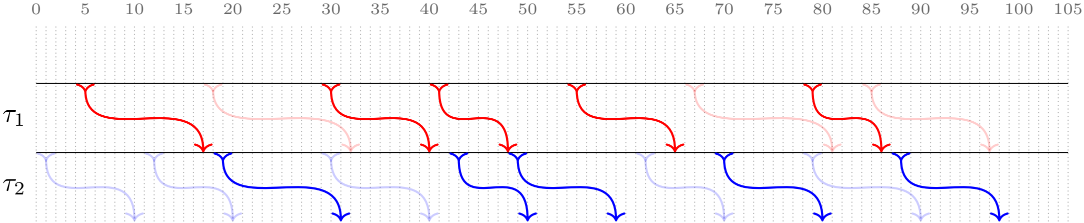
Chain of tasks: analysis





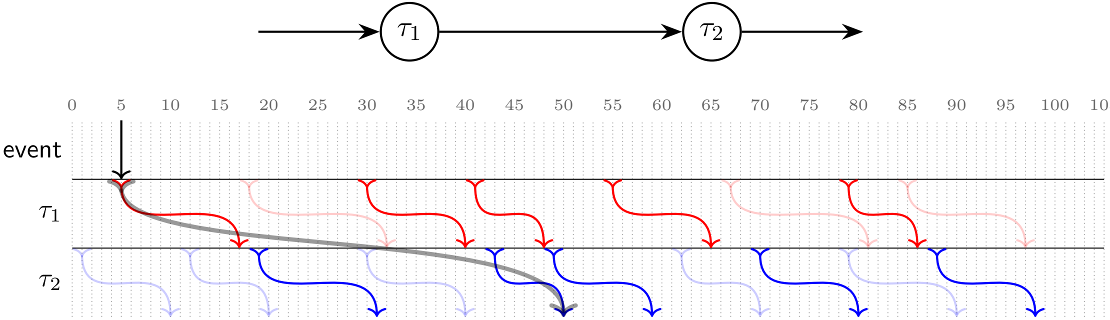
Chain of tasks: analysis





- We can drop "useless" jobs. A job is useless when
  - it writes data being overwritten by next job of the same task, or when
  - it reads data already read by the previous job of the same task

Chain of tasks: analysis



- We can drop "useless" jobs. A job is useless when
  - it writes data being overwritten by next job of the same task, or when
  - it reads data already read by the previous job of the same task
- Typical metric: reaction time (45 in the figure)
  - longest time from an external event (at the begin of the chain) to the end
  - other metrics exist (First-to-Last latency, data age, Last-First, etc.), all related to one another
- Useless jobs do not influence the reaction time

Outline

1 Background

2 Model

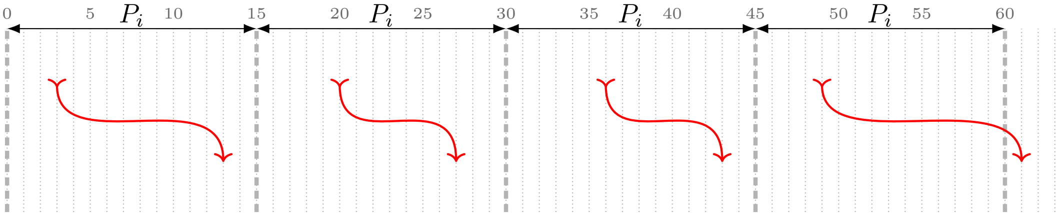
3 Composition

4 Evaluation

5 Conclusion

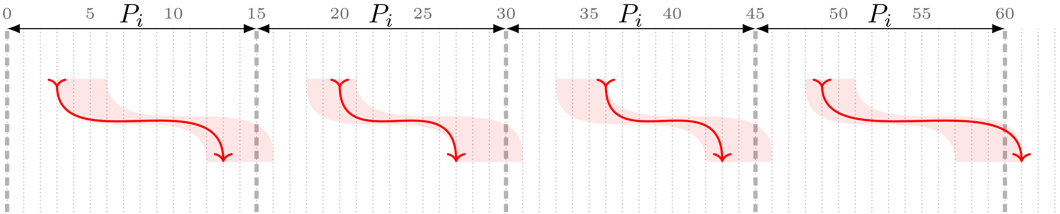
Task model with uncertain read/write events

- Read/write instants may be subject to uncertainties (exec. times, bus contention, etc.)



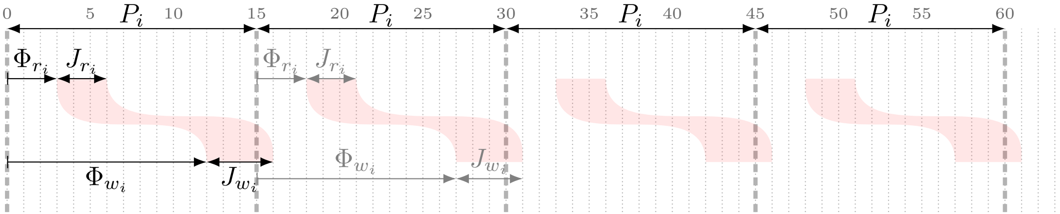
Task model with uncertain read/write events

- Read/write instants may be subject to uncertainties (exec. times, bus contention, etc.)



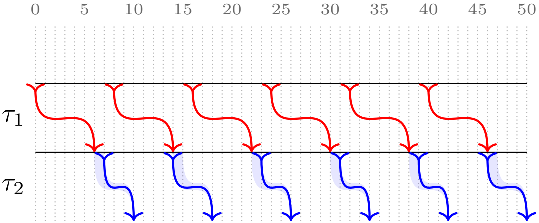
Task model with uncertain read/write events

- Read/write instants may be subject to uncertainties (exec. times, bus contention, etc.)

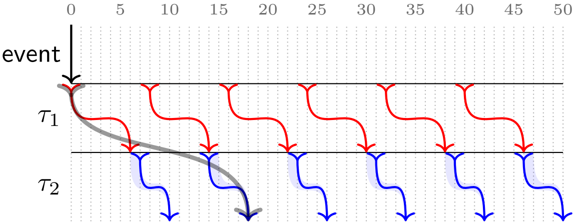


- P_i, *period* of task τ_i
- Φ_{r_i}, *read offset* of τ_i, relative to periodic releases
- J_{r_i}, *read jitter*
- Φ_{w_i}, *write offset* of τ_i relative to periodic releases
- J_{w_i}, *write jitter*
- Intervals are considered closed on both ends

Impact of uncertainty on reaction time

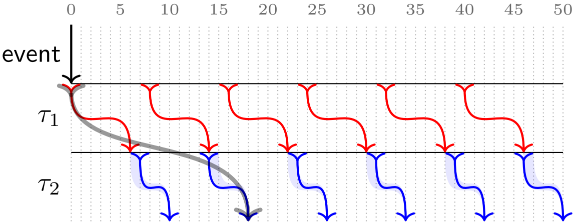


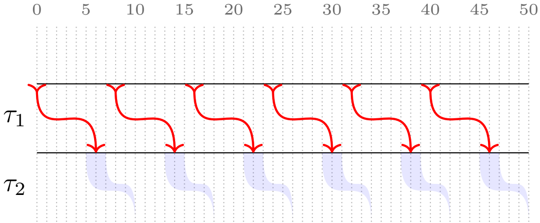
Impact of uncertainty on reaction time



reaction time $= 18$

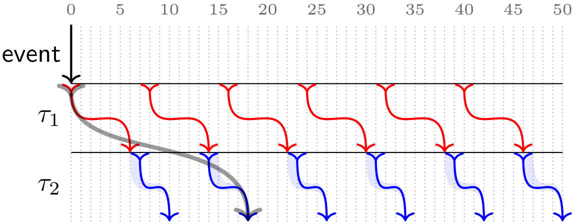
Impact of uncertainty on reaction time

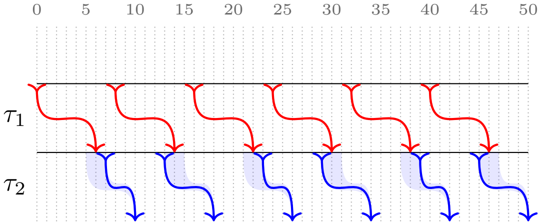




reaction time $= 18$

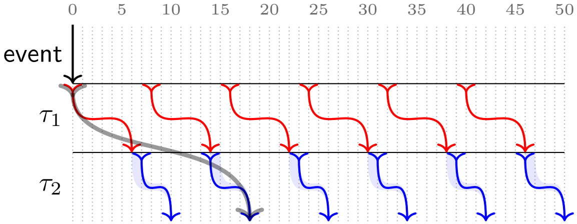
Impact of uncertainty on reaction time



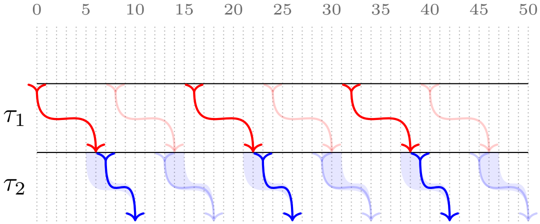


reaction time $= 18$

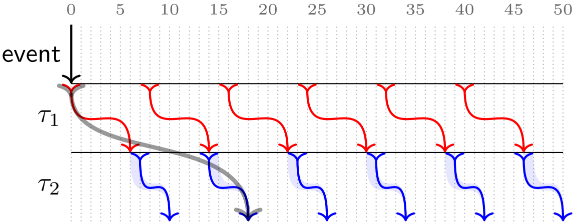
Impact of uncertainty on reaction time



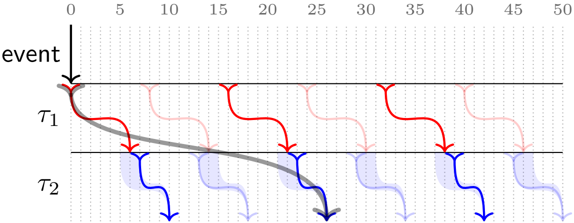
reaction time $= 18$



Impact of uncertainty on reaction time

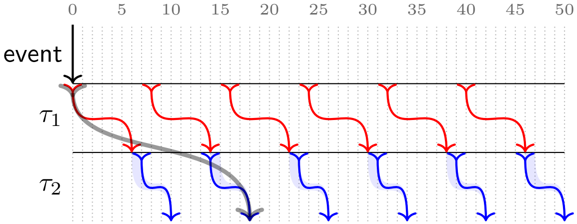


reaction time $= 18$

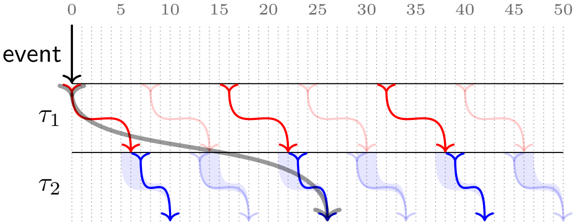


reaction time $= 26$

Impact of uncertainty on reaction time



reaction time $= 18$



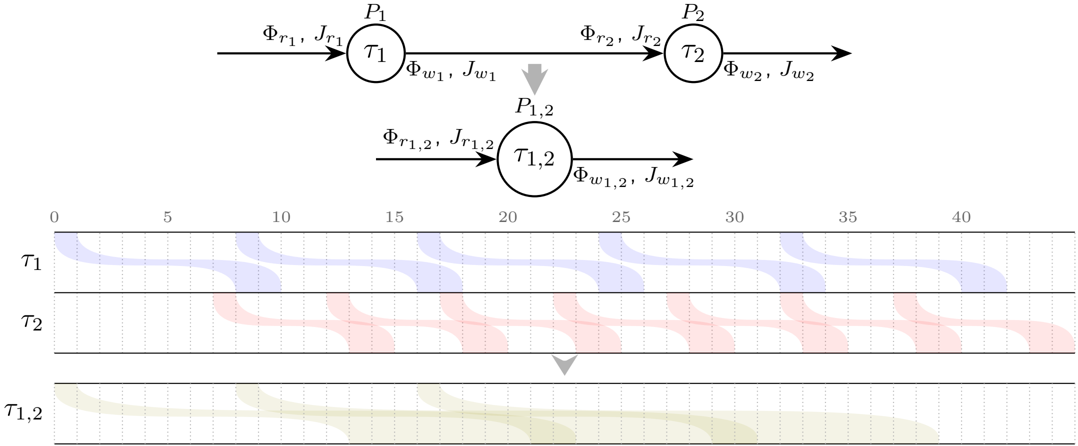
reaction time $= 26$

- small increase in jitter $(+\epsilon)$ makes big jump in reaction time $(+8)$
- worst-case scenario of read instants for max reaction-time is not so intuitive

Outline

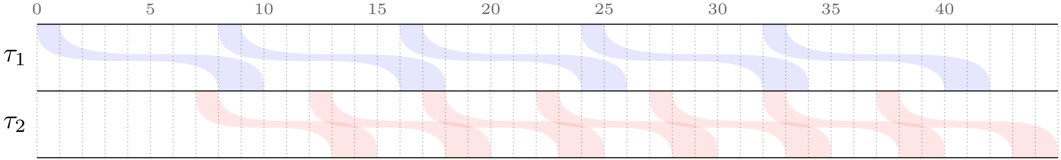
1. Background
2. Model
3. Composition
4. Evaluation
5. Conclusion

Our proposed analysis: composing two tasks into one

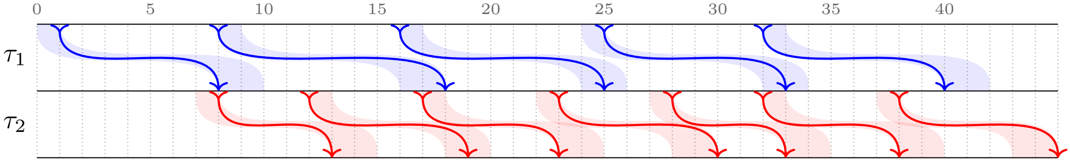


- If we can combine two tasks into one, recursively, we can combine n tasks into one
- Analysis in linear time, instead of spanning all jobs over the hyperperiod
- Reaction time of one task has a closed expression

Composing two tasks: how?

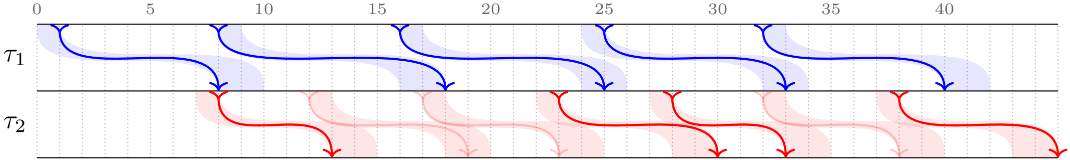


Composing two tasks: how?



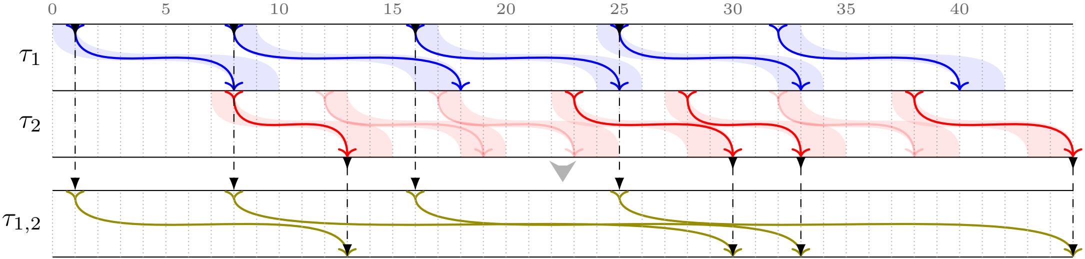
1. **For all sequences** (t_{r_1}, t_{w_1}) and (t_{r_2}, t_{w_2}) compatible with τ_1 and τ_2, respectively

Composing two tasks: how?



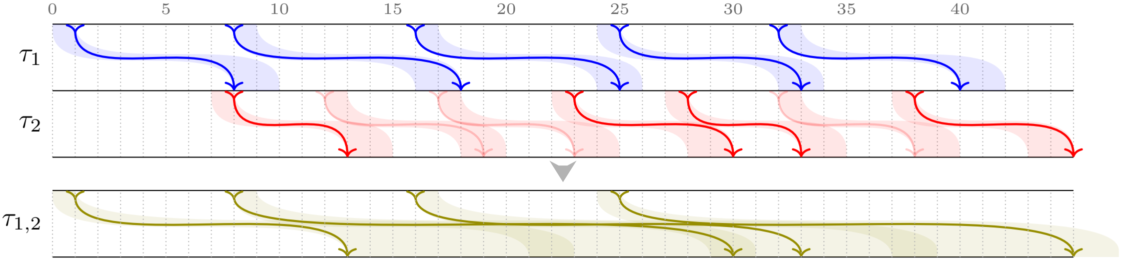
1. **For all sequences** (t_{r_1}, t_{w_1}) and (t_{r_2}, t_{w_2}) compatible with τ_1 and τ_2, respectively
2. We drop the useless jobs

Composing two tasks: how?



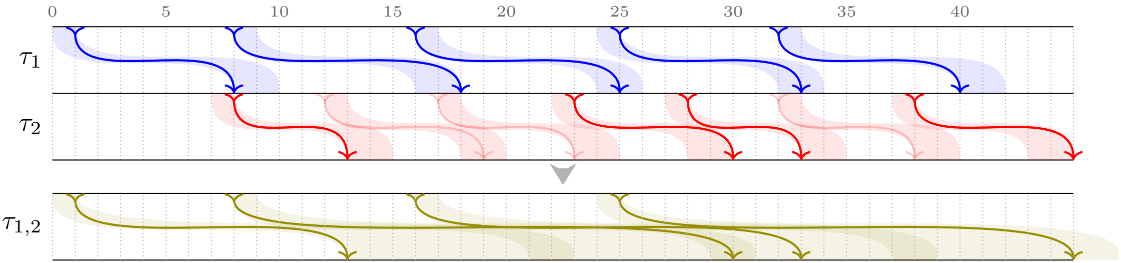
1. **For all sequences** (t_{r_1}, t_{w_1}) and (t_{r_2}, t_{w_2}) compatible with τ_1 and τ_2, respectively
2. We drop the useless jobs
3. We combine the sequences into $(t_{r_{1,2}}, t_{w_{1,2}})$ with read/write from t_{r_1}/t_{w_2}, resp.

Composing two tasks: how?



1. **For all sequences** (t_{r_1}, t_{w_1}) and (t_{r_2}, t_{w_2}) compatible with τ_1 and τ_2, respectively
2. We drop the useless jobs
3. We combine the sequences into $(t_{r_{1,2}}, t_{w_{1,2}})$ with read/write from t_{r_1}/t_{w_2}, resp.
4. We require the sequences $(t_{r_{1,2}}, t_{w_{1,2}})$ be compatible with $\tau_{1,2}$

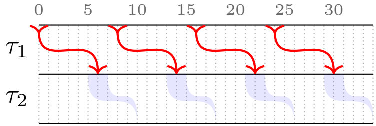
Composing two tasks: how?



1. **For all sequences** (t_{r_1}, t_{w_1}) and (t_{r_2}, t_{w_2}) compatible with τ_1 and τ_2, respectively
2. We drop the useless jobs
3. We combine the sequences into $(t_{r_{1,2}}, t_{w_{1,2}})$ with read/write from t_{r_1}/t_{w_2}, resp.
4. We require the sequences $(t_{r_{1,2}}, t_{w_{1,2}})$ be compatible with $\tau_{1,2}$

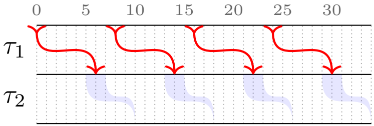
- No recipe for $\tau_{1,2}$, only the needed properties
  - Does $\tau_{1,2}$ always exist? If yes, is it unique?

Composing two tasks: not always possible...



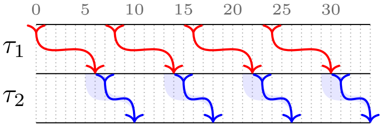
- We need to consider **all** sequences compatible with τ_1 and τ_2

Composing two tasks: not always possible...

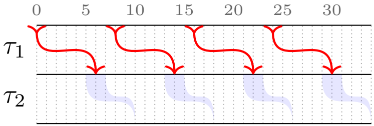


- We need to consider **all** sequences compatible with τ_1 and τ_2

Sequence “late”

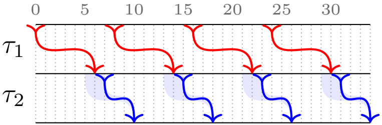


Composing two tasks: not always possible...

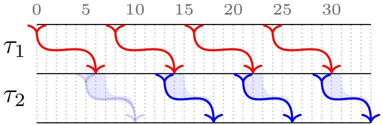


- We need to consider **all** sequences compatible with τ_1 and τ_2

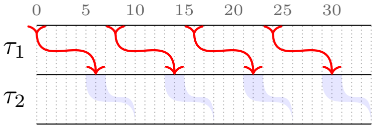
Sequence “late”



Sequence “early”

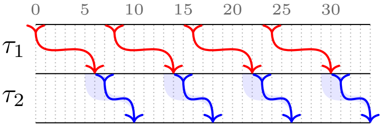


Composing two tasks: not always possible...

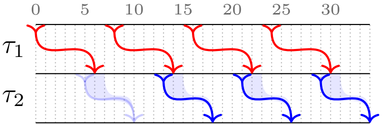


- We need to consider **all** sequences compatible with τ_1 and τ_2

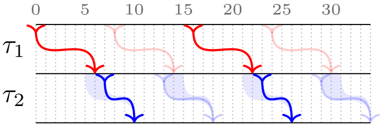
Sequence "late"



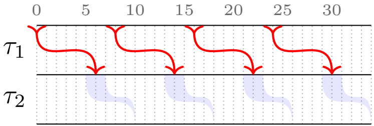
Sequence "early"



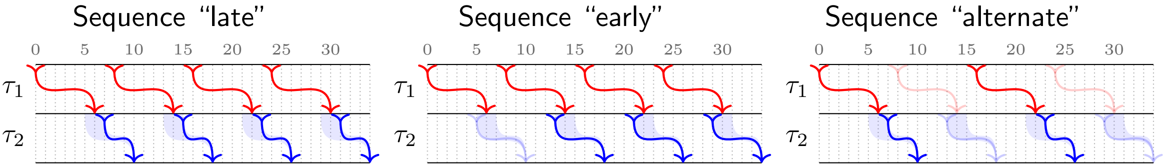
Sequence "alternate"



Composing two tasks: not always possible...

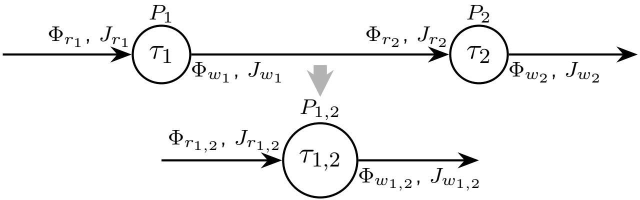


- We need to consider **all** sequences compatible with τ_1 and τ_2



- The period $P_{1,2}$ of $\tau_{1,2}$ needs to be equal to P_1 and equal to $2P_1$: impossible!
- A periodic $\tau_{1,2}$ which can embrace all possible sequences **may not exist**

How to resolve this existential problem?



- **[3 theorems]** "*If the following hypotheses hold*

$$\begin{cases} J_{w_1} \leqslant \operatorname{mod}(\Phi_{r_2} - \Phi_{w_1}, P_2) < P_2 - J_{r_2} & P_1 = P_2 \\ P_2 + J_{r_2} \leqslant P_1 - J_{w_1} & P_1 > P_2 \\ P_1 + J_{w_1} \leqslant P_2 - J_{r_2} & P_1 < P_2 \end{cases}$$

then $\tau_{1,2}$ *exists and its parameters are* **[something]**"

- The conditions are independent of J_{r_1} and J_{w_2}
- $J_{w_1} = J_{r_2} = 0 \Rightarrow$ conditions true. As J_{w_1}, J_{r_2} grow, the conditions get more stringent
- **[something]**: $P_{1,2} = \max\{P_1, P_2\}$, $J_{r_{1,2}} \geqslant J_{r_1}$, $J_{w_{1,2}} \geqslant J_{w_2}$

Outline

1 Background

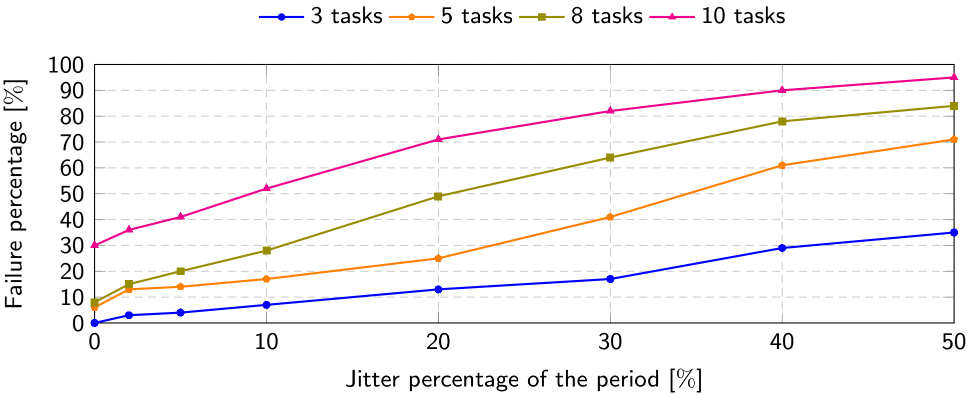
2 Model

3 Composition

4 Evaluation

5 Conclusion

Failure percentage: how often the hypotheses do not hold?



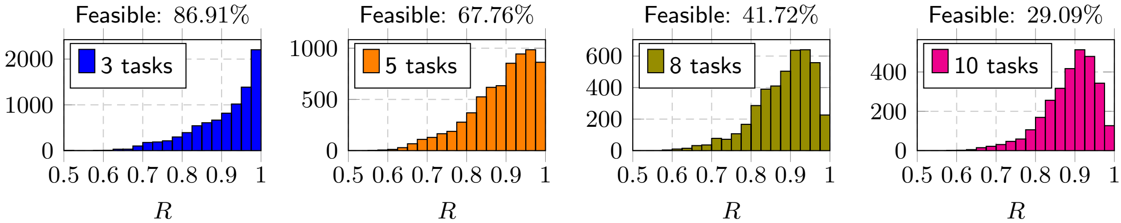
- Growing chain length and jitter reduces the applicability of composition

Quality of the bound

- After composing the n tasks into $\tau_{1,n}$, we find the maximum reaction time as

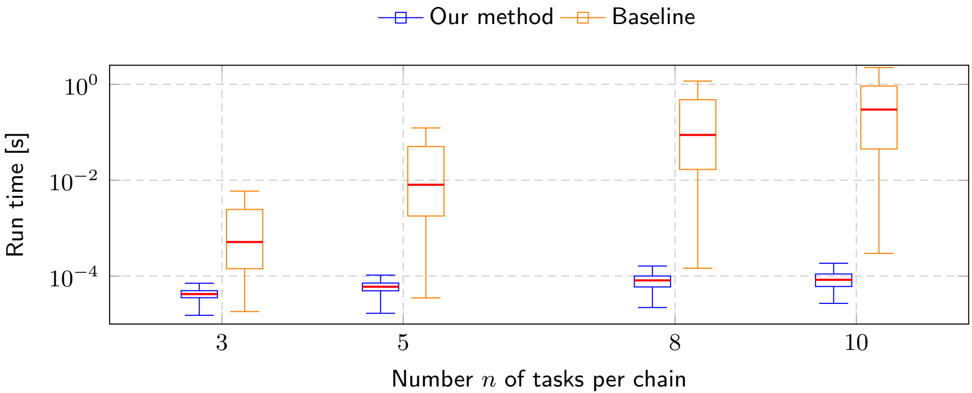
$$P_{1,n} + \Phi_{w_{1,n}} - \Phi_{r_{1,n}} + J_{w_{1,n}}$$

- As baseline for comparison, we use the maximum reaction time by Günzel et al.[2]
- R = baseline maximum reaction time / our method maximum reaction time



[2] M. Günzel, K.-H. Chen, N. Ueter, G. von der Brüggen, M. Dürr, and J.-J̈. Chen, "Timing analysis of asynchronized distributed causeeffect chains," in Real Time and Embedded Technology and Applications Symposium (RTAS), 2021.

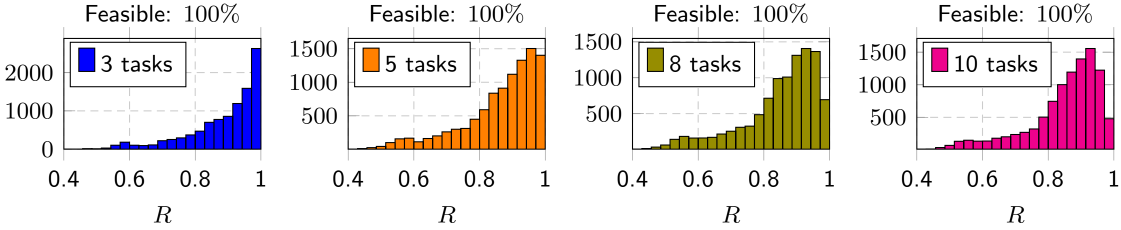
Running time



- Running time is several orders of magnitude smaller than the baseline
    - our composition has linear time, faster than spanning all jobs over the hyperperiod

Active analysis

- When the conditions are violated: "*active*" *analysis*
  - inserting publisher tasks [Bini et al. IEEE-TC'23] between τ_1 and τ_2 to eliminates the jitter



Outline

1. Background
2. Model
3. Composition
4. Evaluation
5. Conclusion

Conclusions & future work

- We propose an **new task model** which incorporates the uncertainty in the read/write instants
- A technique (linear-time) to **compose tasks** and then analysed the composed task only
- Active analysis: if jitter too large to compose, reduce it with publisher tasks
- Can the task model be changed to enable a more successful composition?
- What if DAGs instead of chains?

Github repository with code for the analysis