Controlability and Makespan Issues with Robot Action Planning and Execution[1]

Matthieu Gallien Félix Ingrand

LAAS-CNRS

IWPSS 2006

[1] Part of this work has been supported by the European Social Fund (ESF).

Introduction

- For a planner, the plan execution raises new questions
  - Will a correct plan be always executable ?
  - How to improve efficiency of the system ?
- Autonomous robotics context
  - Taking time into account (visibility windows, task durations, ...)
  - Uncertain task durations and task effects

Solutions

- Three improvements of an existing system: IxTeT
  - Taking into account temporal uncertainties
  - Optimization of the plan time makespan
  - Improving the plan repair strategy

Plan

Description of IxTeT

- General Presentation
- Planning and Execution
- Dynamic Controlability
- New Heuristic

Experimentation

- Presentation
- Results

Presentation of IxTeT

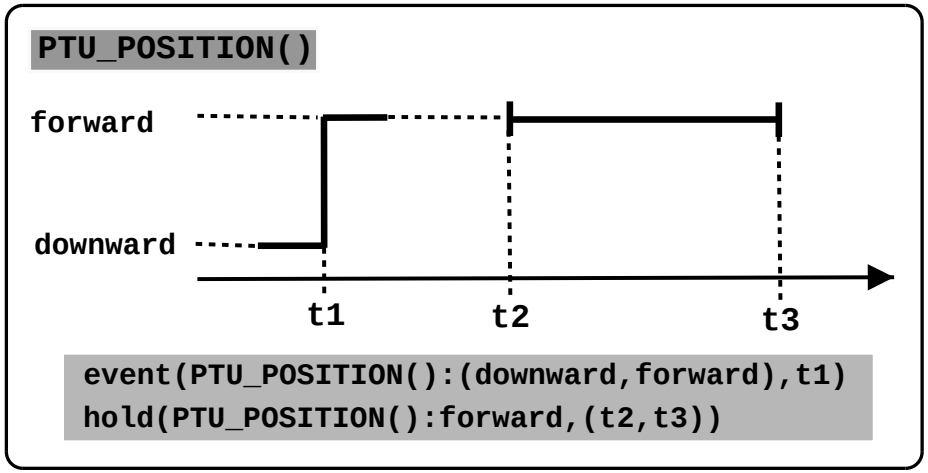
- Plan space planner (POCL)
- Explicit representation of time
- Now embody a temporal executive
- Can partially repair a plan while executing the remaining of the plan
- Consistency of the plan checked with two CSPs: one STN and a general CSP
  - Flexible plans easy to execute and adapt

Typical Mission: Exploration

- Initially unknown environment: duration of navigations is uncertain
- Take pictures, communicate during visibility (visibility of an orbiter, ...)
- The system makes its best to accomplish the mission
- New goals are added, tasks fails, ...
- IxTeT validated using a real robot and its simulator

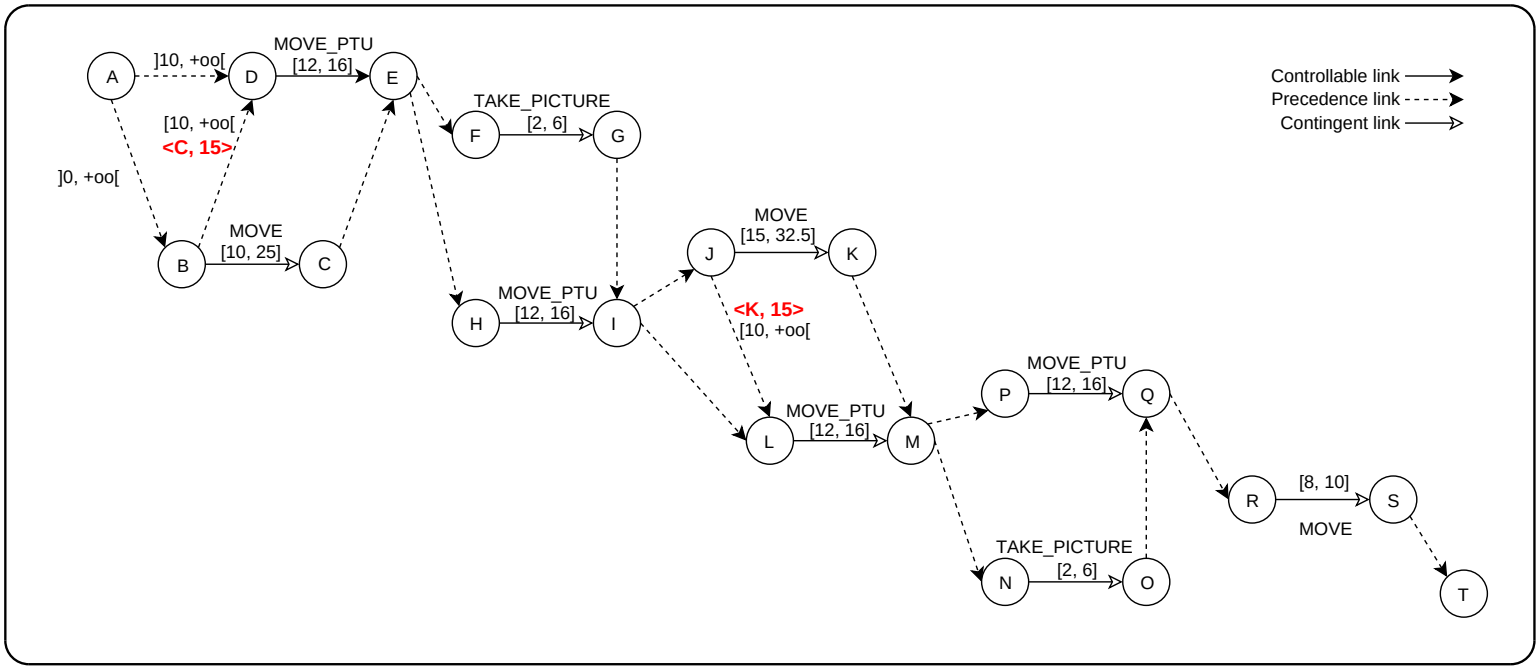
Temporal Planning

- State variables: piecewise constant functions of time



- Variable based representation (temporal and atemporal)
- Able to monitor resource usage
- Task model partially instantiated
- Relevant approaches: EUROPA, IDEA, CASPER

Produced Plan



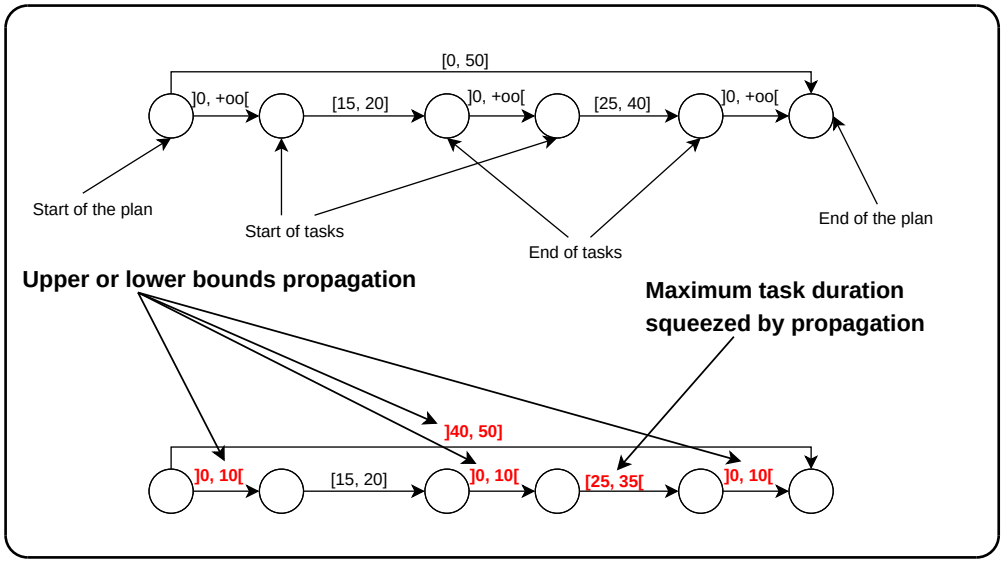
Execution

- Incremental plan execution
- An execution cycle:
  - Sensing new events
  - Plan repair if necessary
  - Executing what is needed
- Allows the plan repair to be distributed over several cycles
- Replanning from scratch if needed (failed plan repair)
  - All running tasks are stopped

The Statement

- How to guarantee a correct execution when faced with temporal uncertainties ?
  - The dynamic controlability statement
- Some actions are controlable
  - IxTeT decides the start and the end time
- Others are not controlable
  - The system starts them
  - It must wait the end of the task by observing it
  - Their durations are not known precisely

Why and How



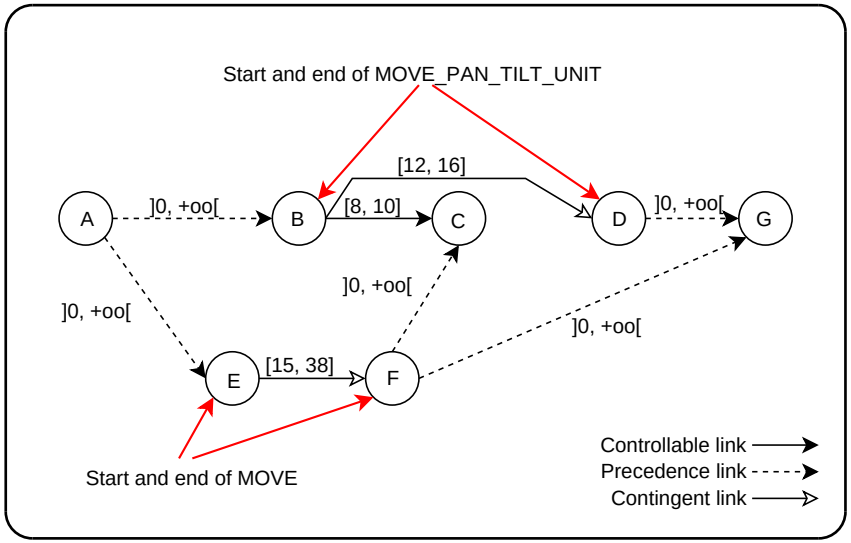
- The durations are flexible constraints like: duration of a tasks anytime between 15 and 20s for example
- Consistency checking by propagation:
  - May squeezed possible durations
    - Squeezing an uncontrollable duration may lead to a failure

Definition

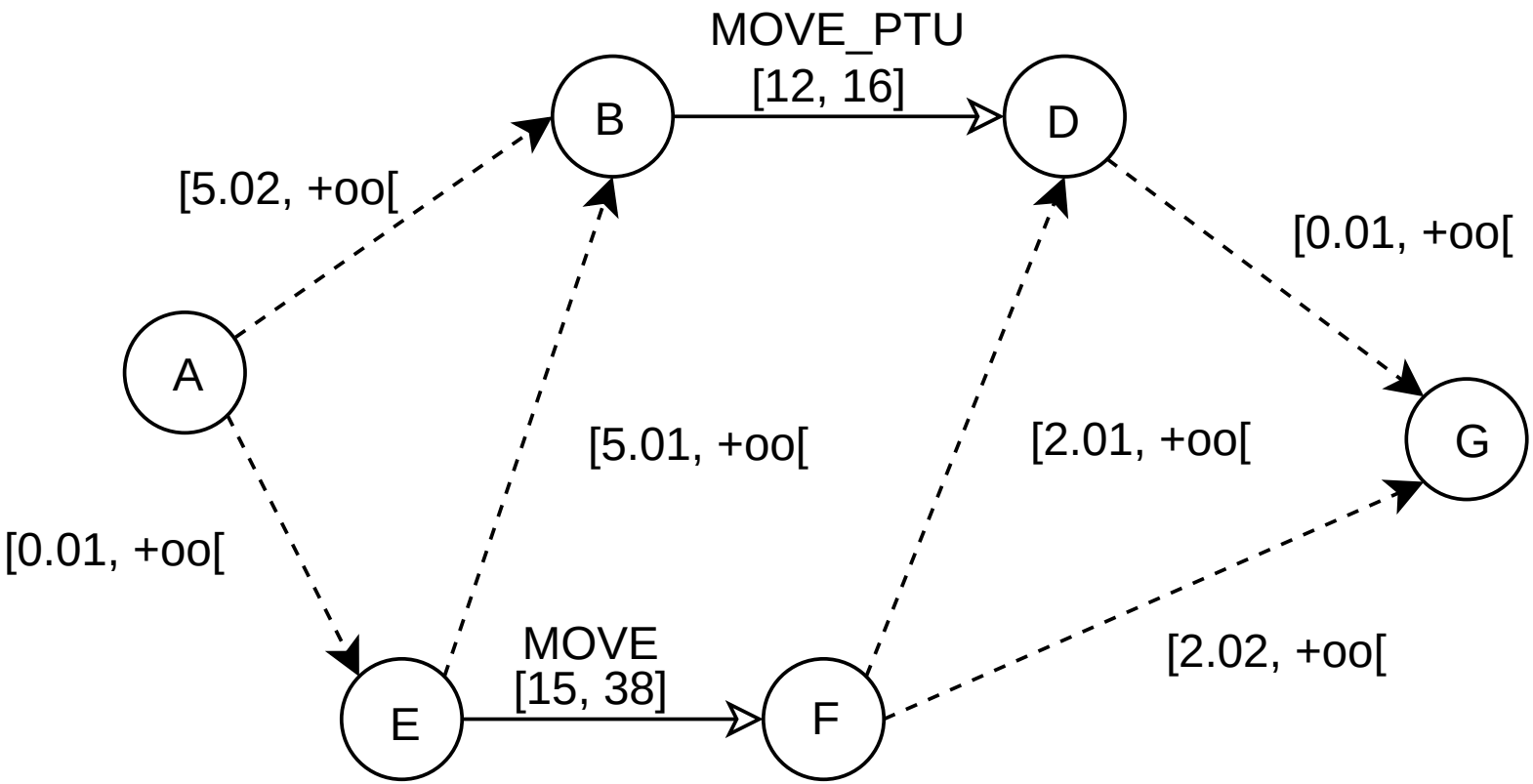
- Classically, temporal planning uses simple temporal network (STN)
  - No uncertainty
- STNU [Vidal et al 97]: new framework with uncertainties
  - Replace consistency by controlabilities
  - Weak controlability
  - Dynamic controlability
  - Strong controlability

Example Definition

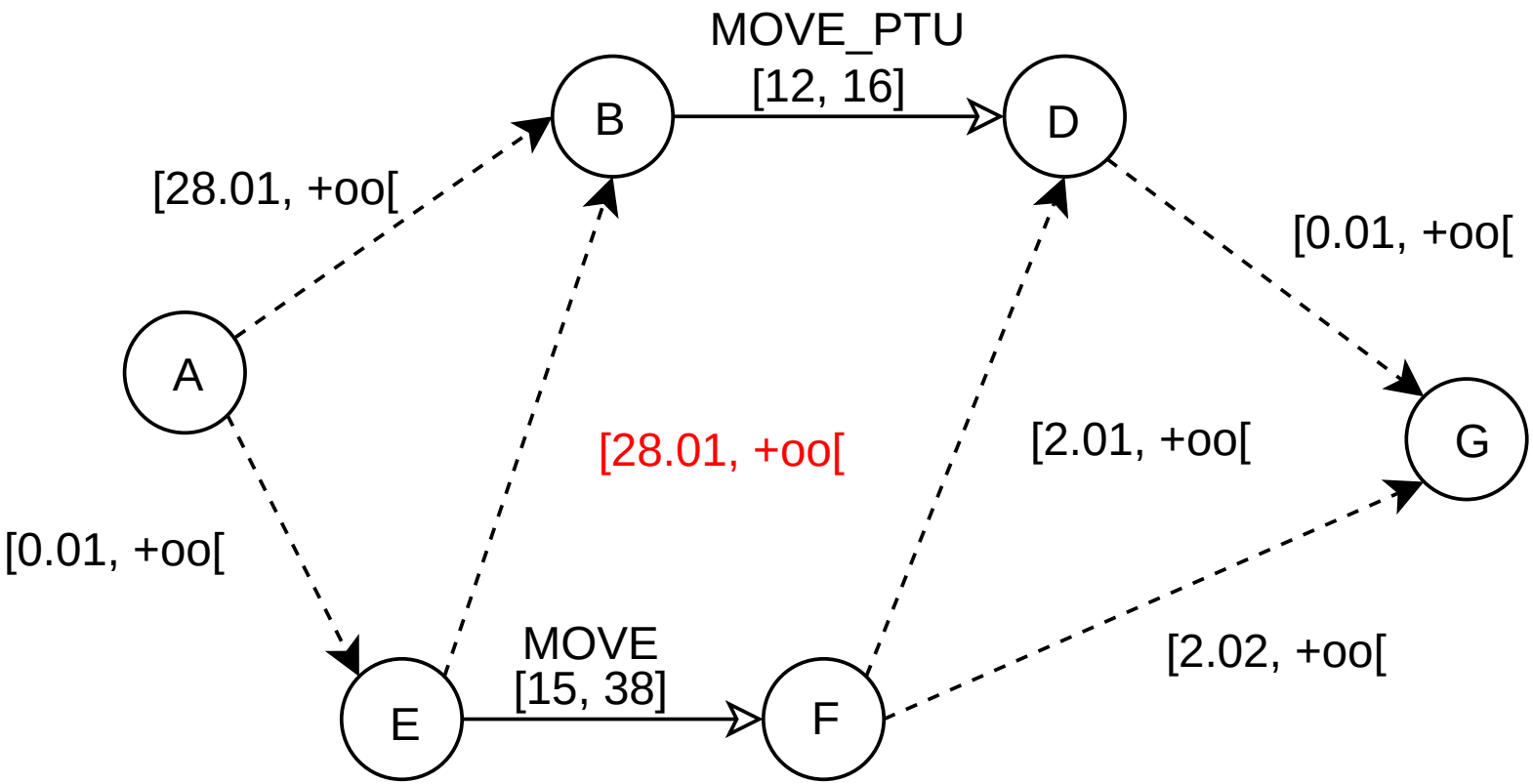
- For the purpose of demonstration, we simulate a need for heating the pan&tilt unit for the camera before moving it
  - The constraints are such than: moving the robot is compatible with moving the cameras
  - For robustness, the heating will allways be done before the move



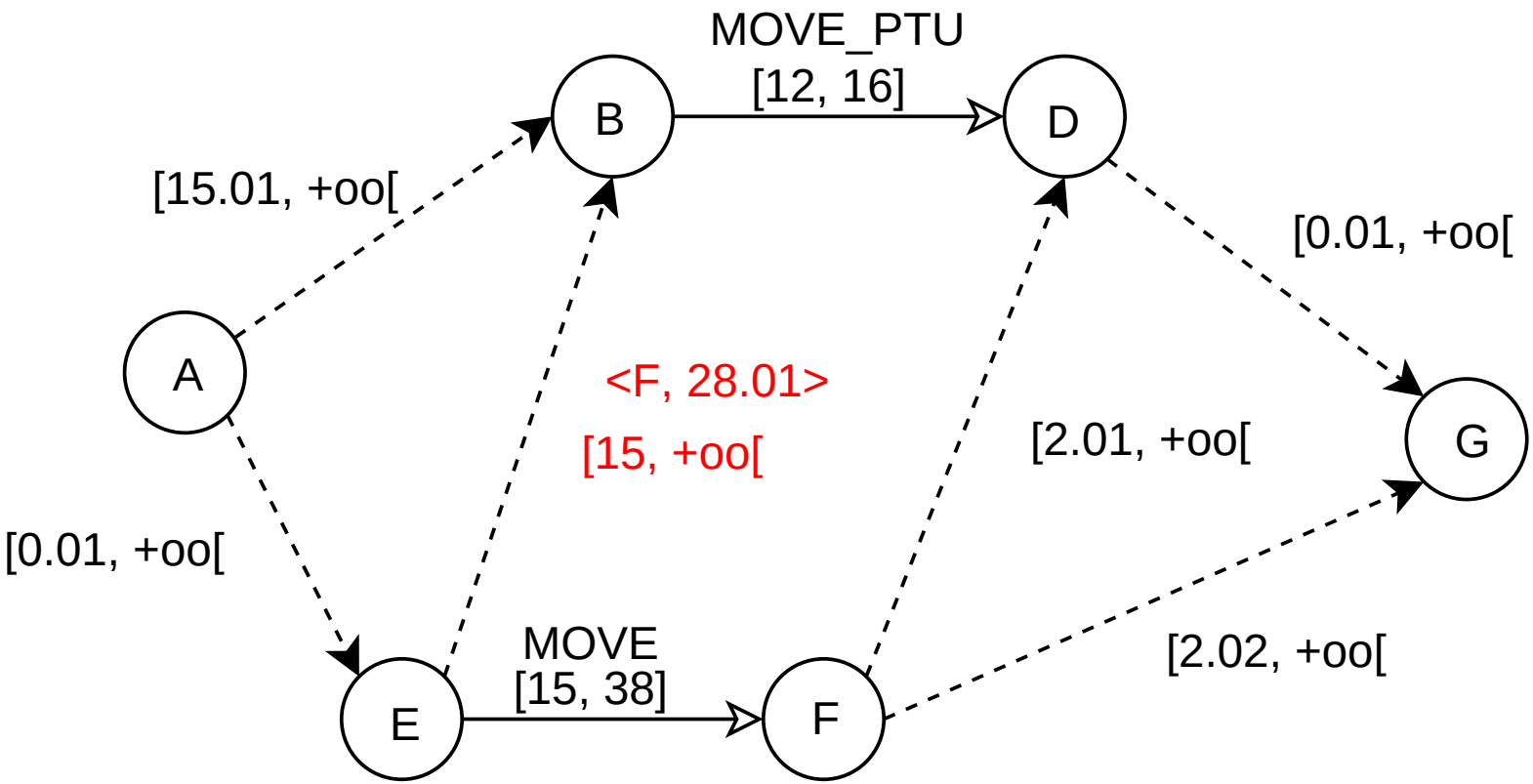
Example of STNU



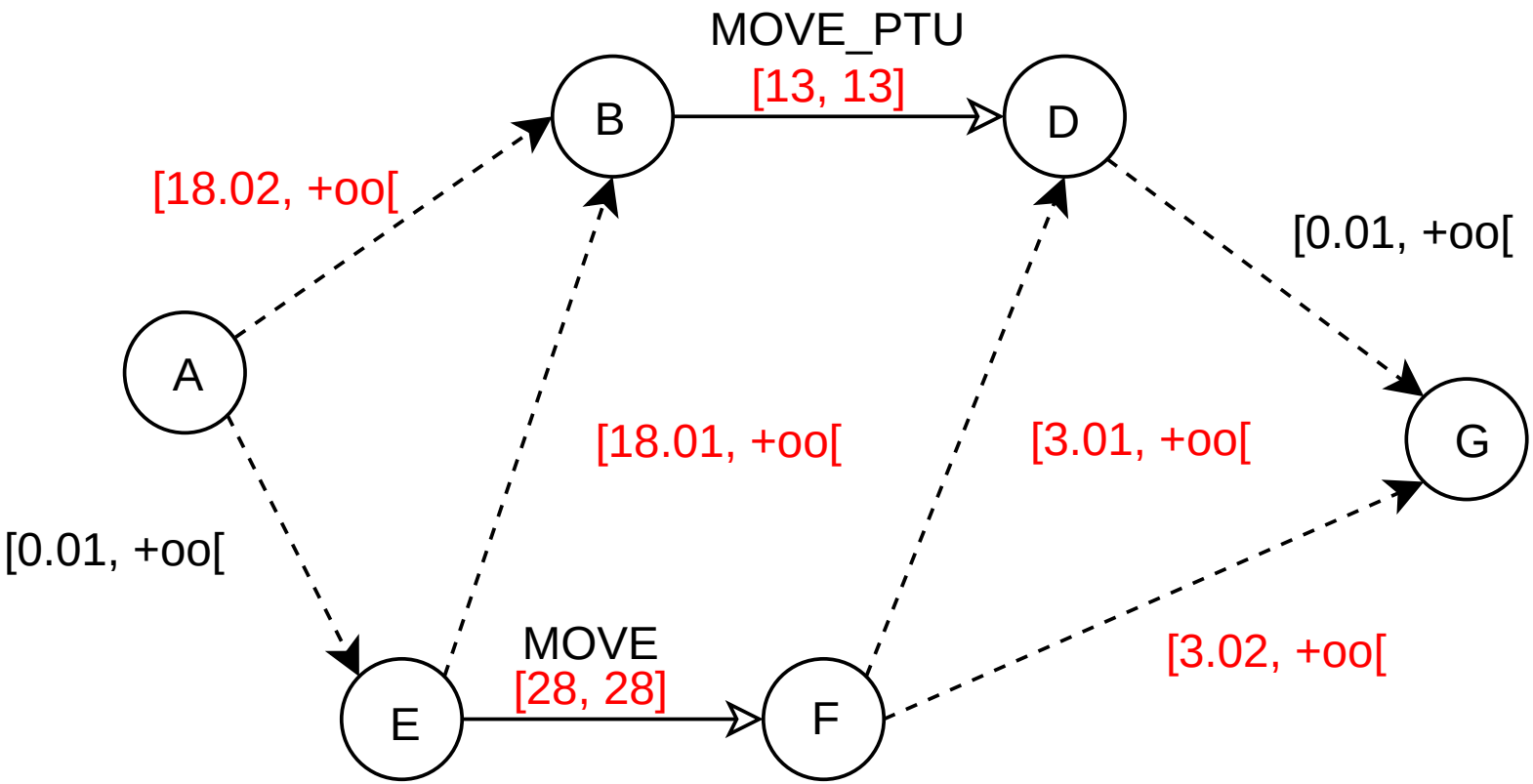
Strong Controlability



Dynamic Controlability

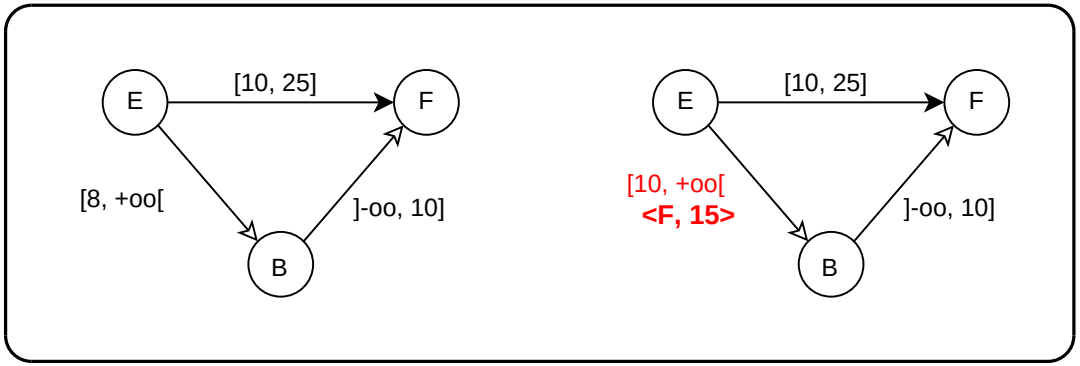


Weak Controlability



Algorithm 3DC+ [Morris et al. 2001]

- Introduce “waits" whose are conditional constraints
  - Complete, polynomial and correct algorithm
- Analyze triangles of constraints
- Require more computations than path consistency of an STN
- Integration of an STNU in IxTeT allows to make contingent plans

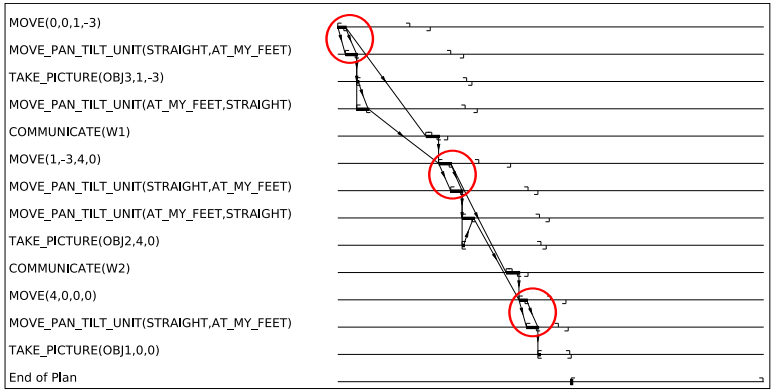


Presentation

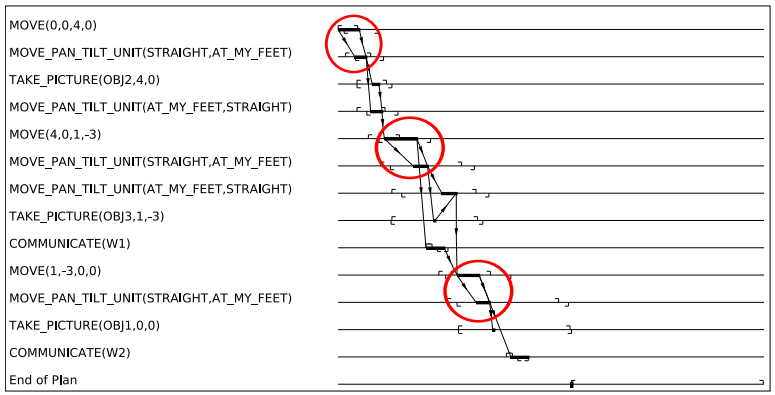
- IxTeT an heuristic planner
  - Depth first search
- Least commitment heuristic
  - Depend on a computed cost for each resolvant
  - Theoretically the cost depends on the number of removed solutions
  - In practice, it is computed with one formula by resolvant type

How

- The classical heuristic makes an equal repartition over the whole plan of all tasks:

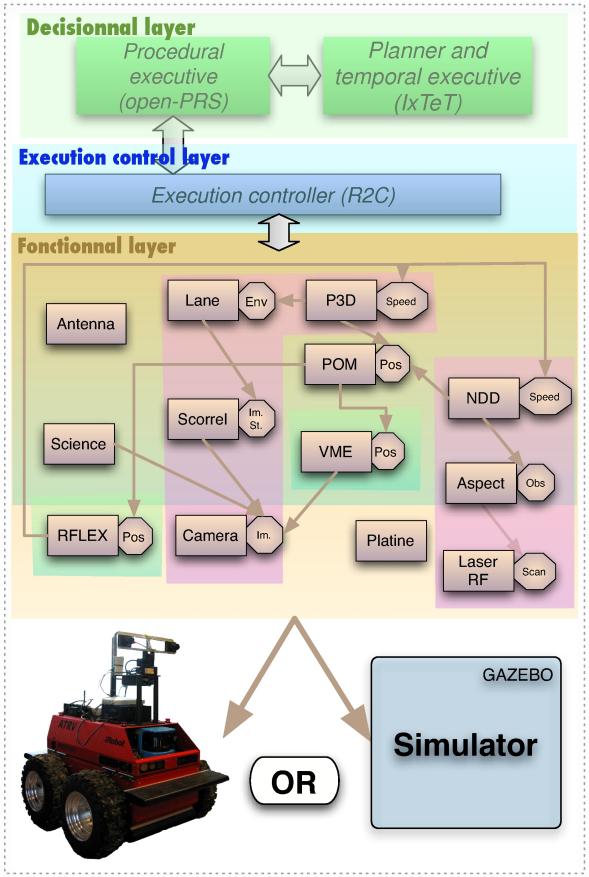


- The new one modify the temporal related cost functions to try to minimize the makespan



Plan

The LAAS Architecture



- IxTeT has been integrated in the LAAS architecture
- It works on a real robot and on its simulator

Result concerning the new heuristic

- Most of initial plans are shorter, some are not better
- Plan robustness:
  - Not good if used with an STN
  - Good if used with an STNU
- Not a significant difference on planning duration

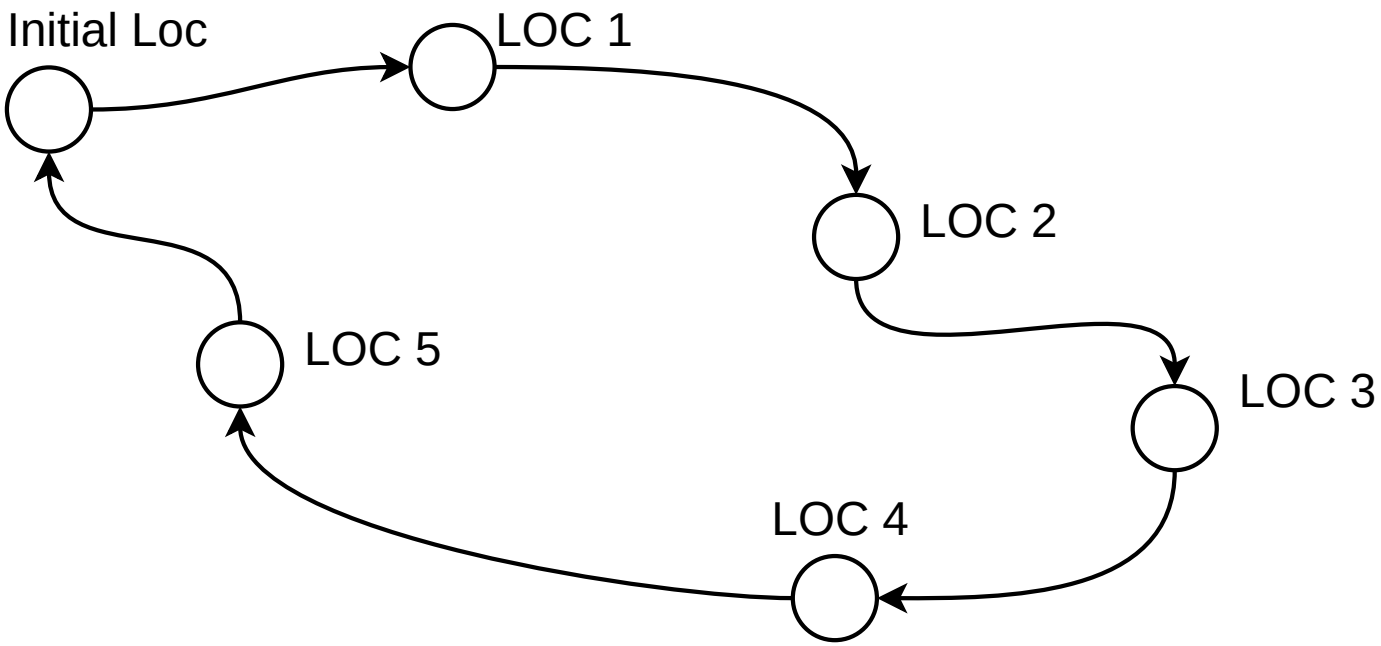
Results With an STN

- May work if you are lucky: statistically 50% of goals are abandonned
- May have very bad effect:
  - May produce trivially not executable plans, instead of cancelling low priority goals and fails to execute the more priority goals

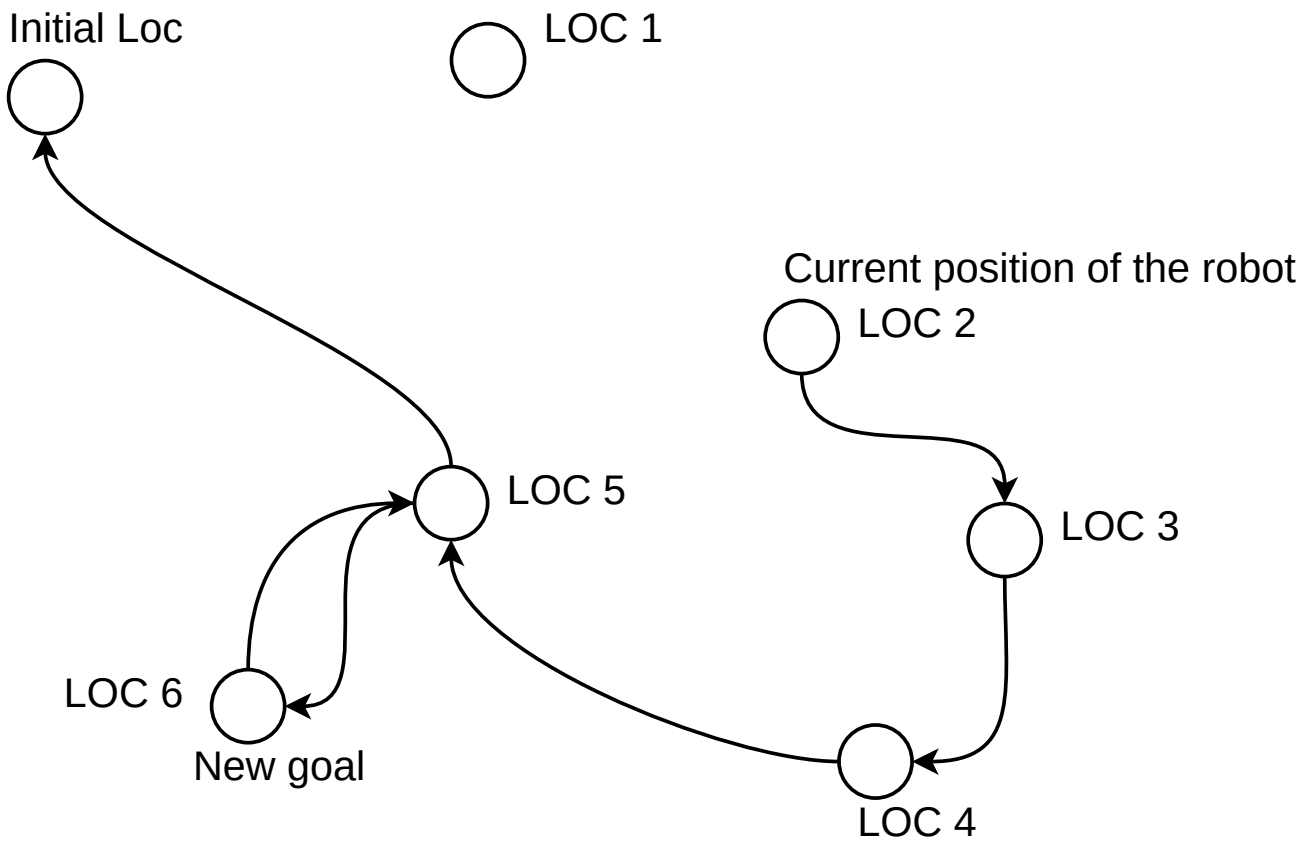
Better Results with an STNU

- In all runs except one, all goals were achieved
- At execution, it takes 10ms for each timepoint, with an STN it takes 1ms
- The planning duration with 8 goals is 3 times more than with an STN

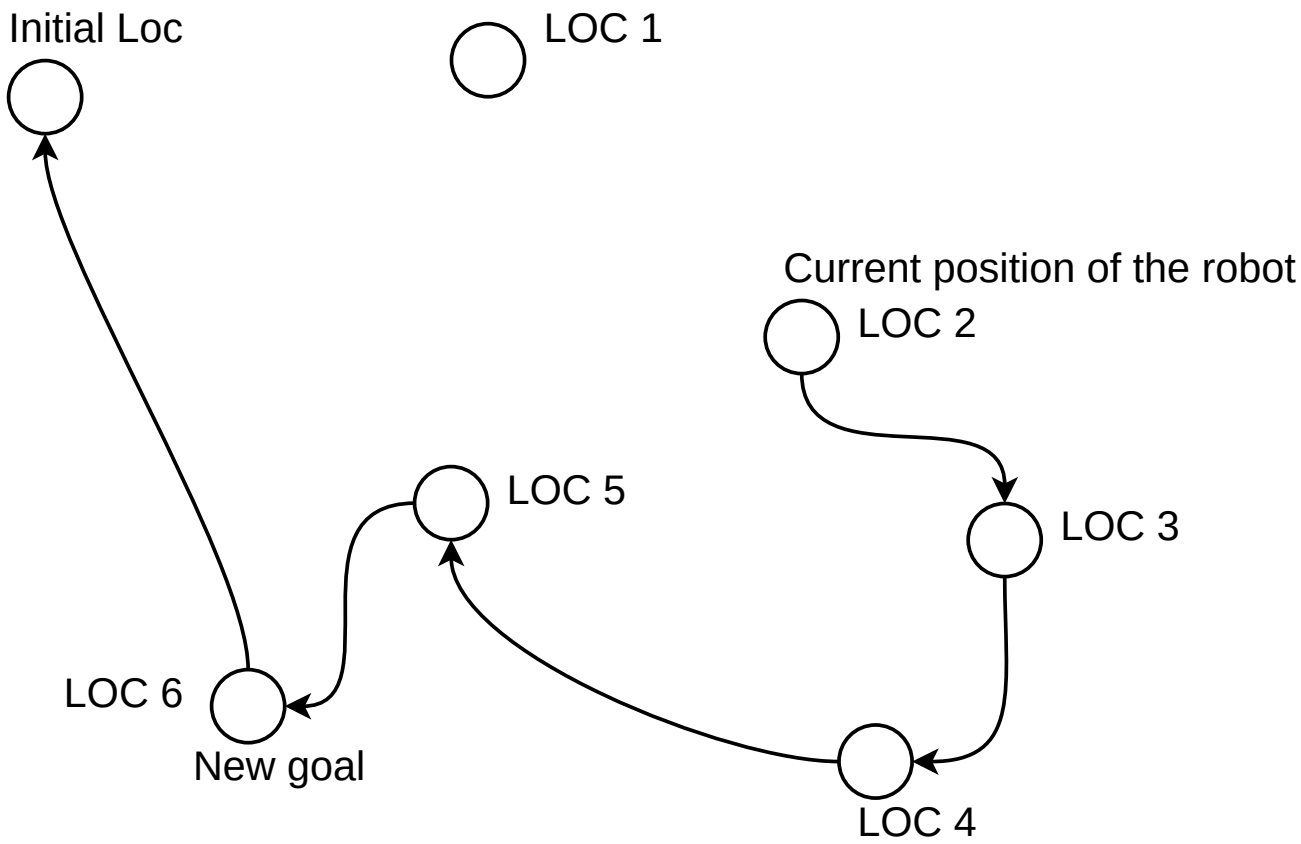
Example of Plan Repair



Inefficiency of Actual Plan Repair



The New Plan Repair



Conclusion

- IxTeT a system for plan generation and execution
  - New problematics due to plan execution
  - Integrated a temporal framework with uncertainties
  - Use a makespan minimizing heuristic
- The simulator, a way to evaluate the system versus other versions
- Current work on upgrading the plan repair algorithm