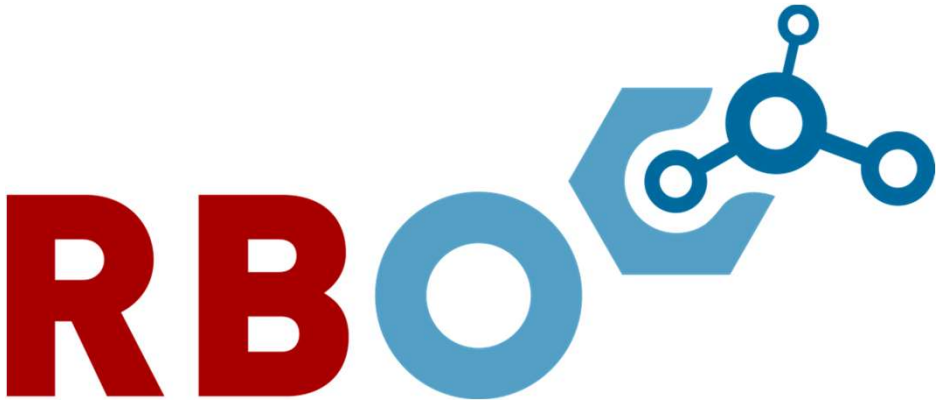


Motion Planning: Coding

Xing Li, xing.li@tu-berlin.de

Technische Universität Berlin

Robotics Library: www.roboticslibrary.org

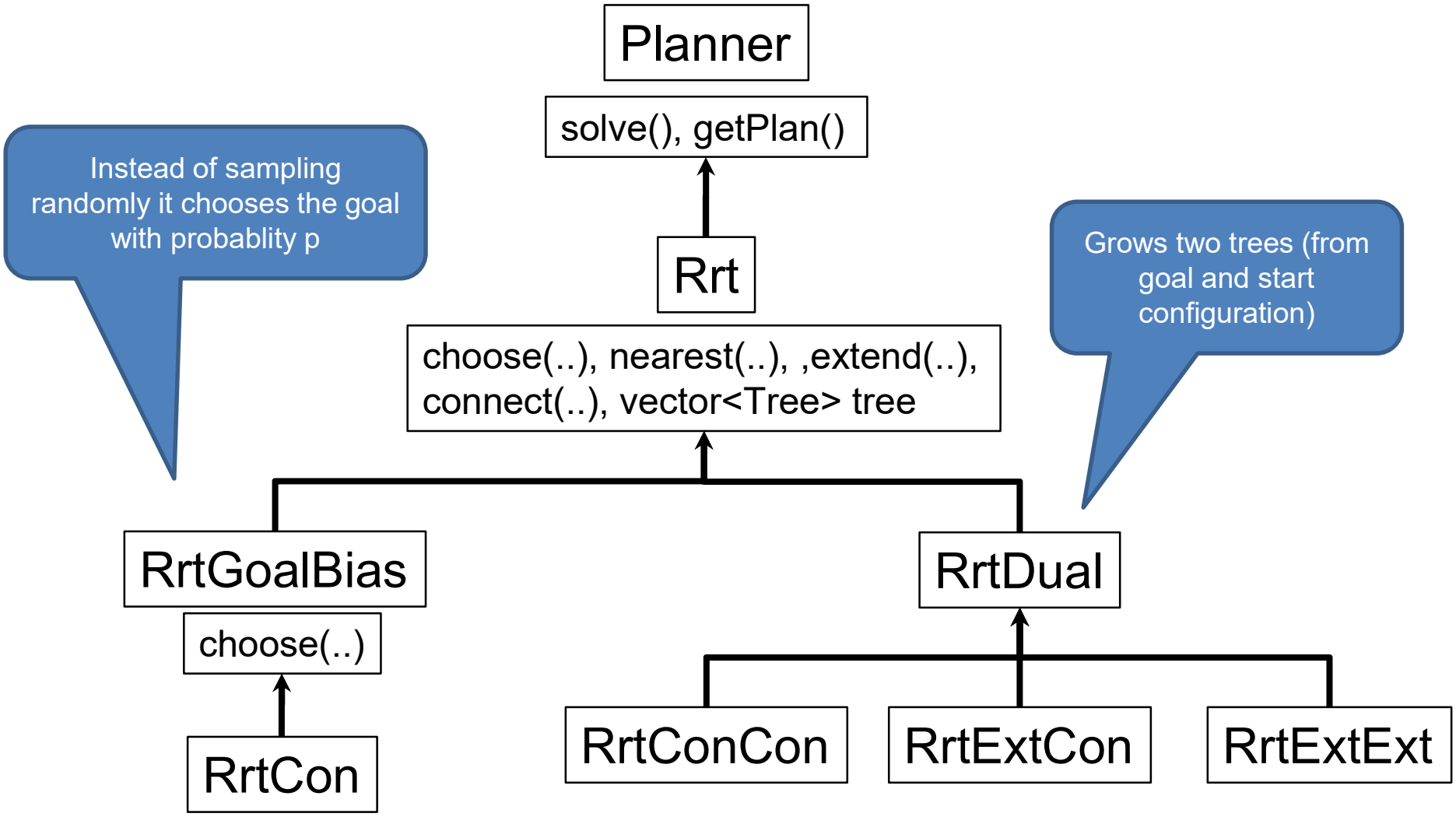
- A framework for developing robotics applications

math Mathematics	**util** Timers, Threads, Mutexes, …	**xml** XML Abstraction
hal Hardware Abstraction	**kin** DH-Kinematics	**mdl** Rigid Body Kinematics/Dynamics
sg Scene Graph Abstraction	**plan** Motion Planning	**ctrl** Operational Space Control

- Pro: All models we need are already included. It also provides a lot of examples.
- Con: Sparse documentation

  http://doc.roboticslibrary.org/0.7.0/index.html
  https://github.com/roboticslibrary/rl/tree/master/src/rl/plan
  https://www.roboticslibrary.org/api

Hierarchy of RRT Classes in rl::plan

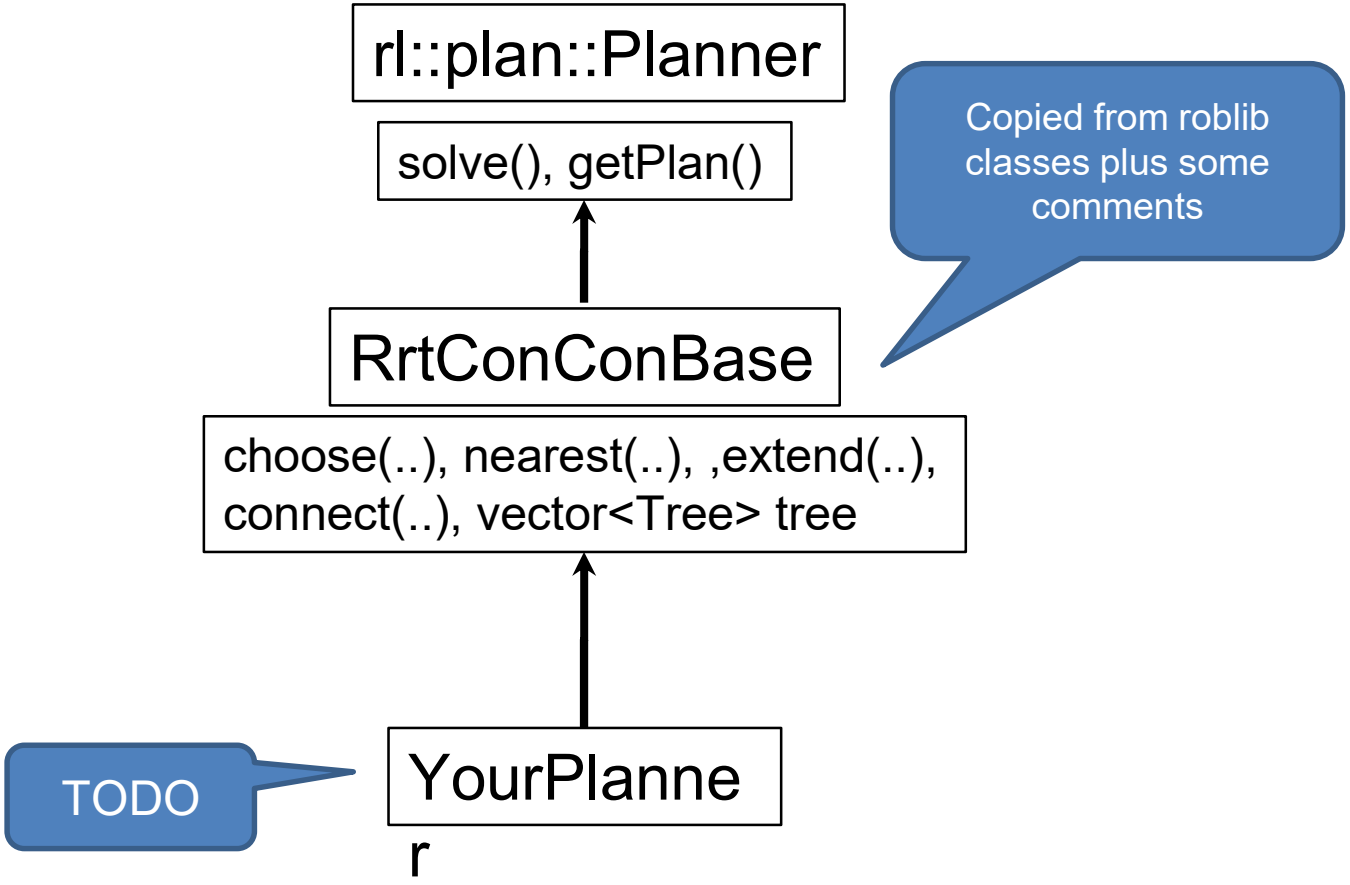


You can easily install RobLib on Ubuntu

- Install prerequisites
  - `sudo add-apt-repository ppa:roblib/ppa`
  - `sudo apt-get update`
  - `sudo apt-get install librl-dev`
  - Download and extract tutorialPlan.zip:
  - `tar xfz tutorialPlan.zip`
- Build
  - `cd tutorialPlan/build`
  - `cmake ..`
  - `make`
- Test:
  - `cd tutorialPlan/build`
  - `./tutorialPlan - Press space to start planning, F12 to reset`

https://www.roboticslibrary.org/tutorials/install-ubuntu/

Hierarchy of RRT Classes for assignment



Writing Your Planner

- Modify YourPlanner.cpp
- Inherit from RrtConConBase (or any other RRT based planner in rl::plan)
- Set parameters of your algorithm in TutorialPlanSystem.cpp
- Implement better choose(), connect(), extend(), solve()
- You can add parameters for vertices in RrtConConBase::VertexBundle

Evaluating your algorithm

- Every time you run `./tutorialPlan` an entry is added to `benchmark.csv`
- This entry shows: computation time needed to solve the problem, number of nodes and edges created, number of queries needed to construct them, etc.
- Run your algorithm at least ten times!

Writing the Documentation

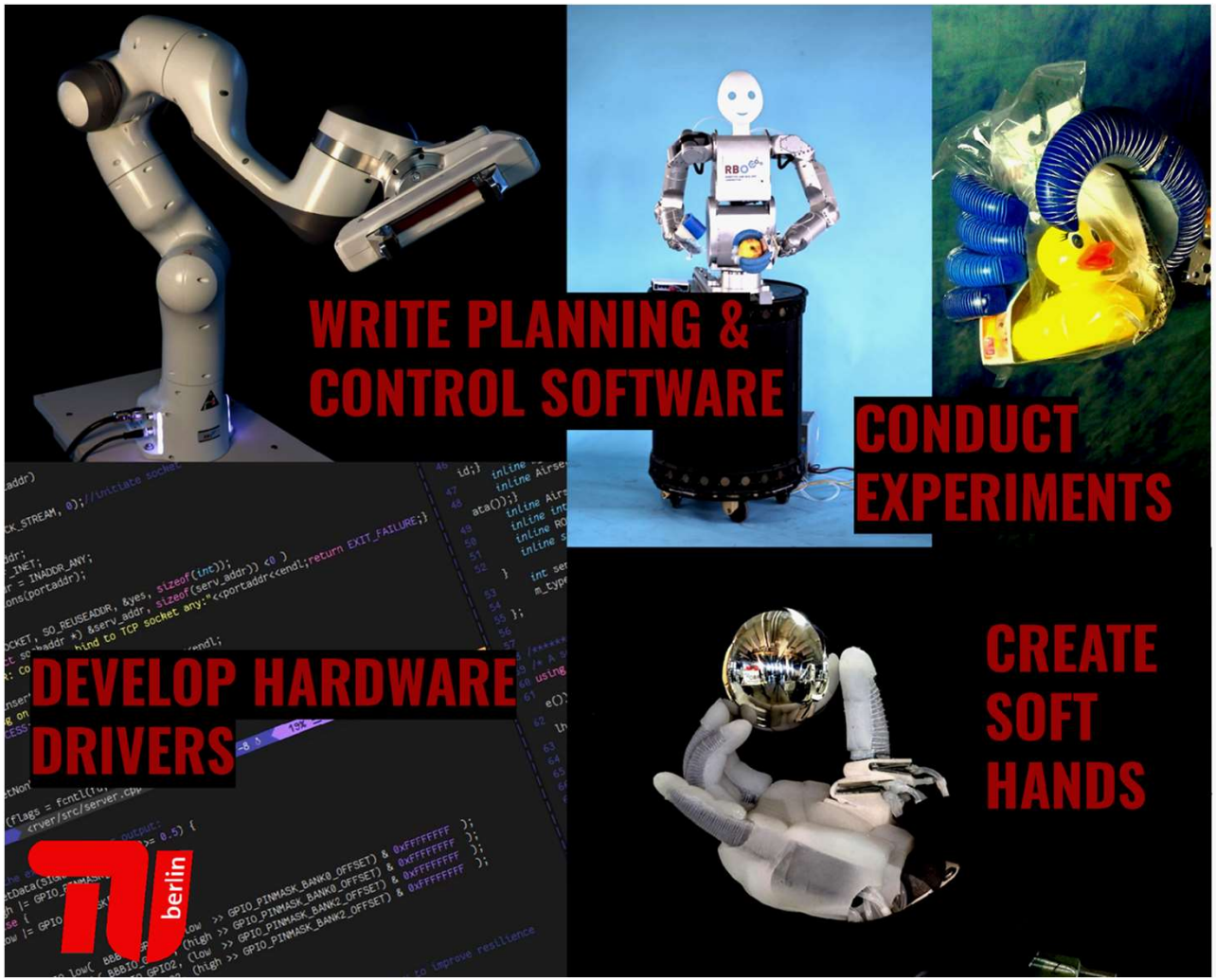
- Explanation of the proposed extensions with a code snipped
- Reasoning about extensions that have not improved the runtime
- Performance evaluation of your final algorithm
- (Optional) Extra points can be gained by performing an ablation study of your extensions with regard to the runtime

References

- http://roboticslibrary.org
- Robotics library examples: https://github.com/roboticslibrary/rl-examples
- Introduction to RRTs: msl.cs.uiuc.edu/rrt/
- Planning Book: Sections in 5.5 and 14.4 in planning.cs.uiuc.edu
- RRTExtCon: http://msl.cs.uiuc.edu/~lavalle/papers/KufLav00.pdf
- Avoid exhausted Nodes: http://homepages.laas.fr/jcortes/Papers/icra07paper.pdf
- Avoid Voronoi bias: http://ieeexplore.ieee.org/iel5/10495/33250/01570709.pdf

RBO

TU berlin



STUDENT ASSISTANTS WANTED

robotics.tu-berlin.de/menue/open_positions

ROBOTICS AND BIOLOGY LABORATORY

Skills:

- C++ / Python
- Linux
- English