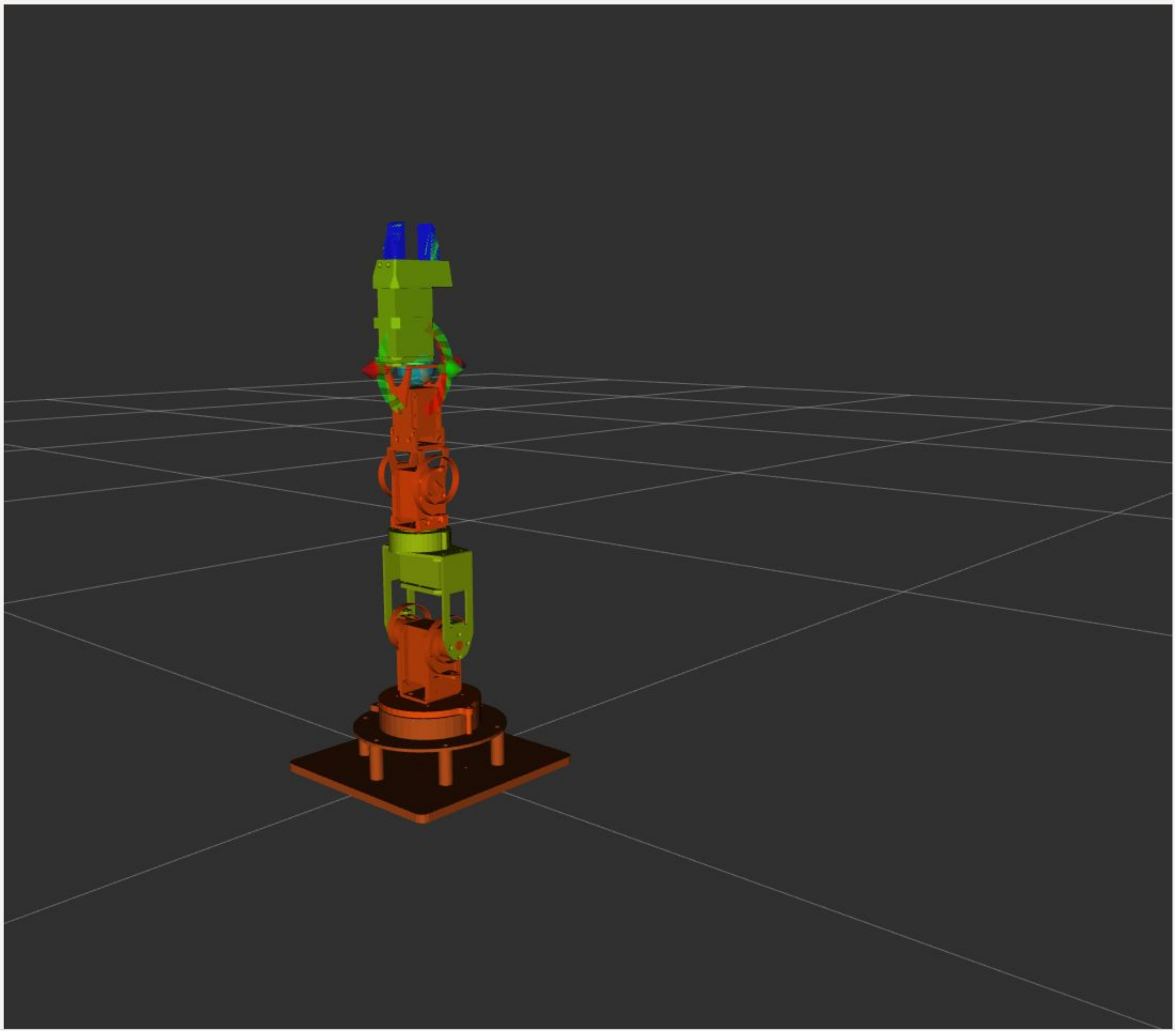
A study of Trajectory planner in move it and implementation with Cool400

Submitted for requirement of the questionnaire provided by:

Asimov Robotics

Mudit Jindal

YMCA University of Science and Technology, Faridabad



COOL 400

1. Introduction

Cool400 is a robotic arm with 6 joints, 6 arm links and 1 base link. The tool in the case of Cool400 is a gripper. A 6D arm with a gripper can be used for pick and place applications.The planners studied in this report are :

(OMPL)
BFMT, BKPIECE, BiEST, BiTRRT, EST, FMT, KPIECE, LBKPIECE, LBTRRT, LazyPRM, LazyPRMstark, PDST, PRM, PRMstark, ProjEST, RRTConnect, RRT, RRTstark, SBL, SPARS, SPARStwo, STRIDE, TRRT

(CHOMP)
CHOMP

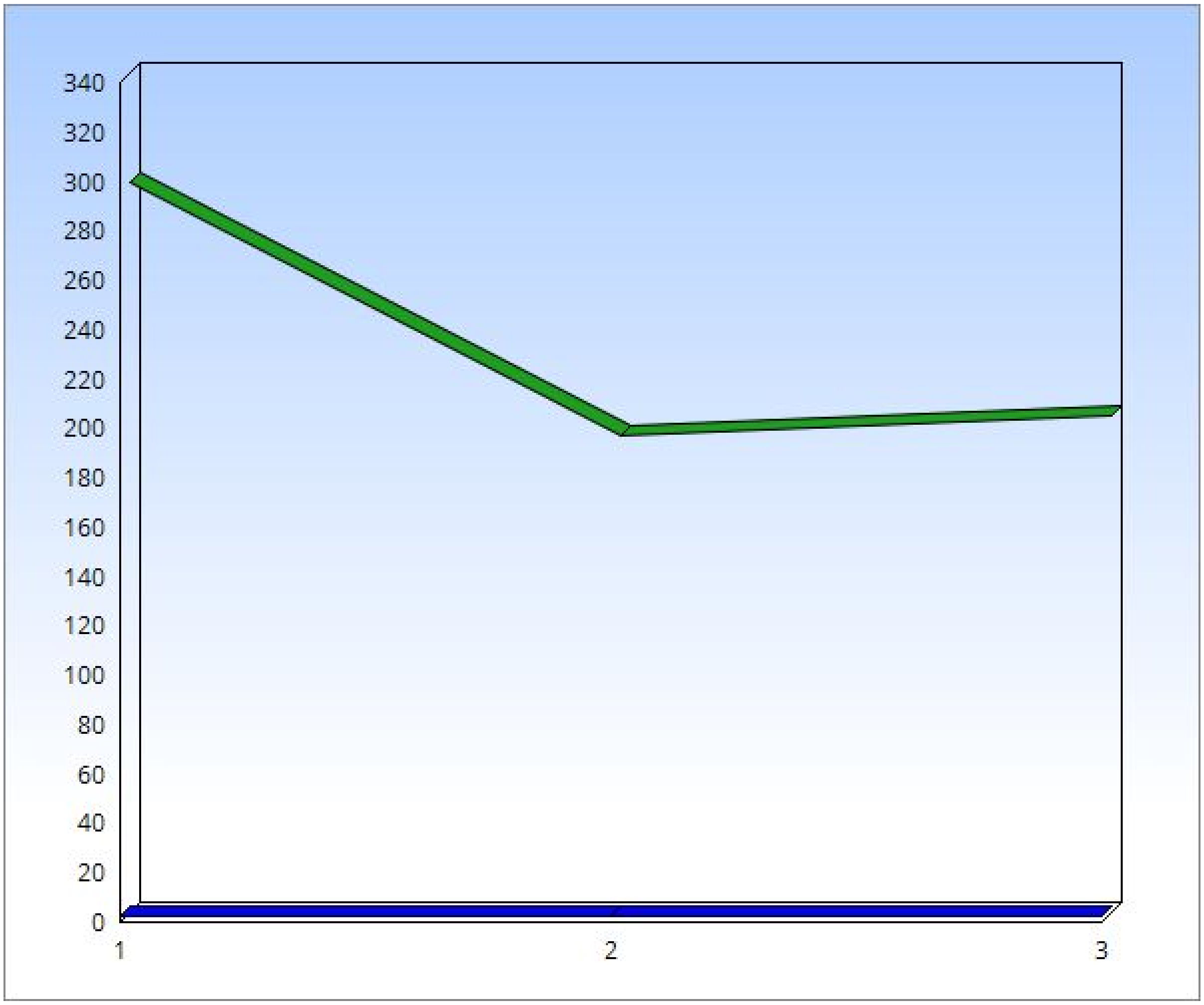
There are other planners such as SBPL and STOMP also available in moveit but are under development and are currently unstable.

2. Performance analysis of OMPL based planners :

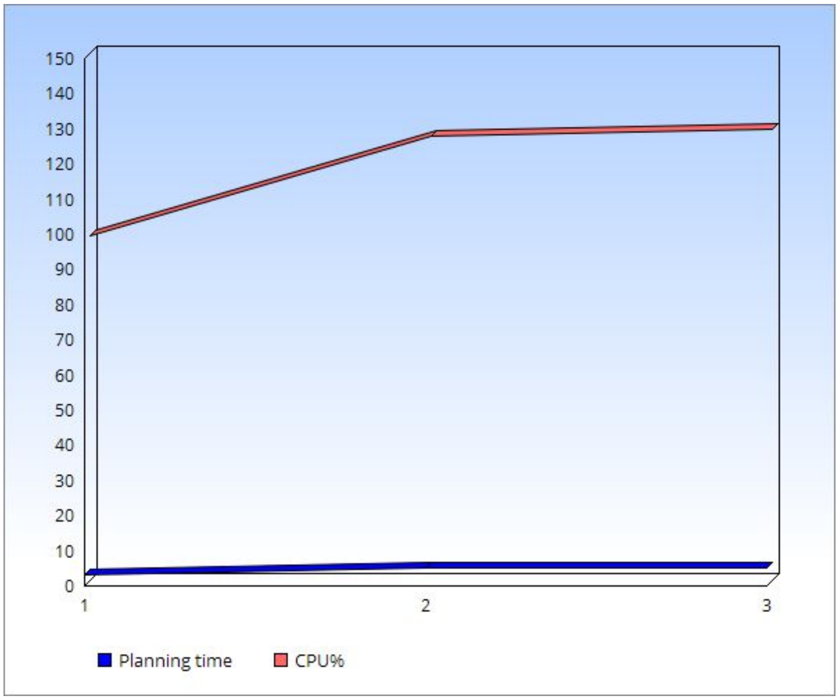
The following tests were performed on all available planners in OMPL and CHOMP. The machine used during the tests is as follows :-

Lenovo Z51- i5 6th gen - 4GB RAM - 2GB ATi graphics with ubuntu 16.04 and ROS kinetic.

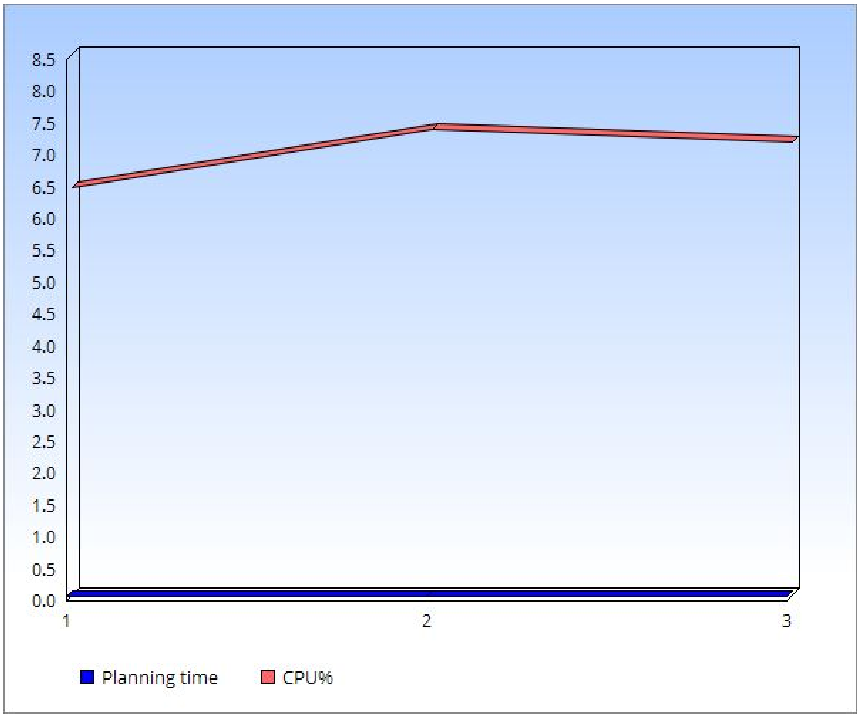
In the following graphs, X-axis represents iteration number and Y-axis represents CPU usage in green/red and time taken in blue.



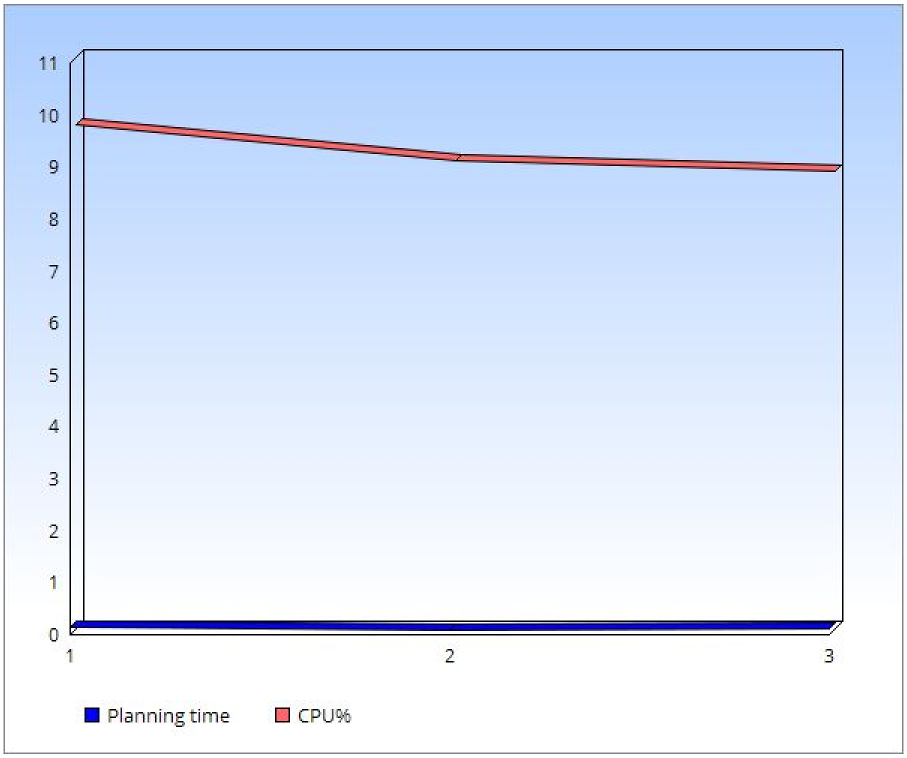
BFMT



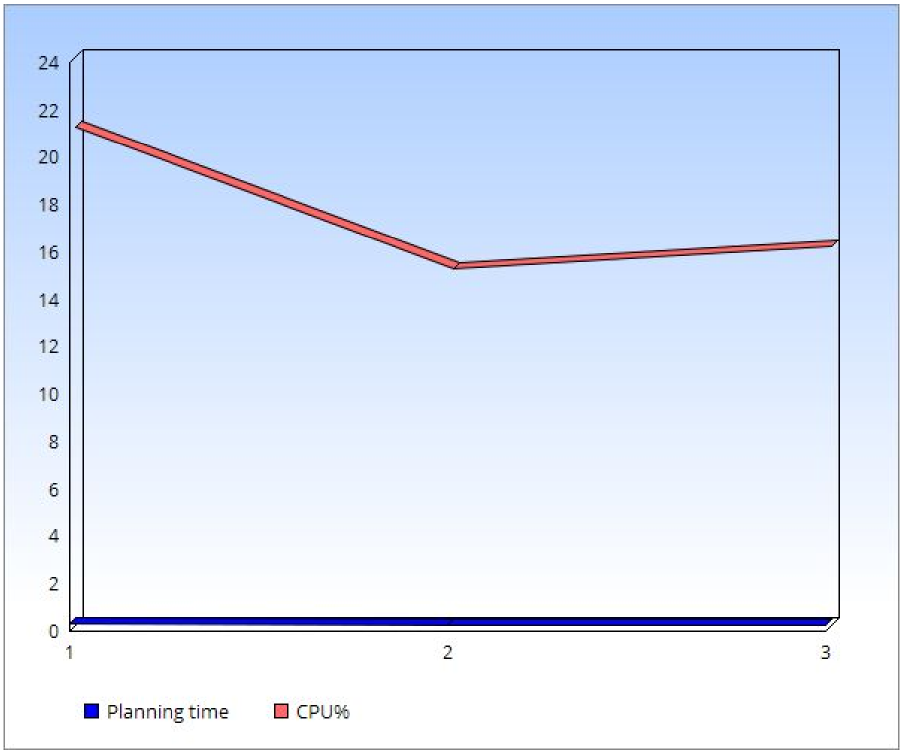
BKPIECE



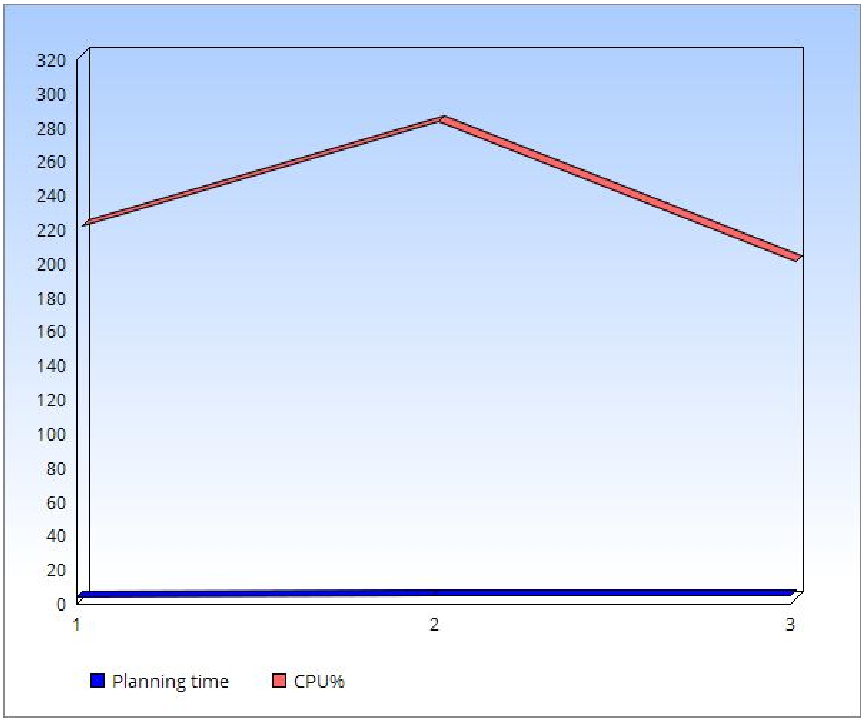
BiEST


11
10
9
8
7
6
5
4
3
2
1
0
1
2
3
Planning time
CPU%

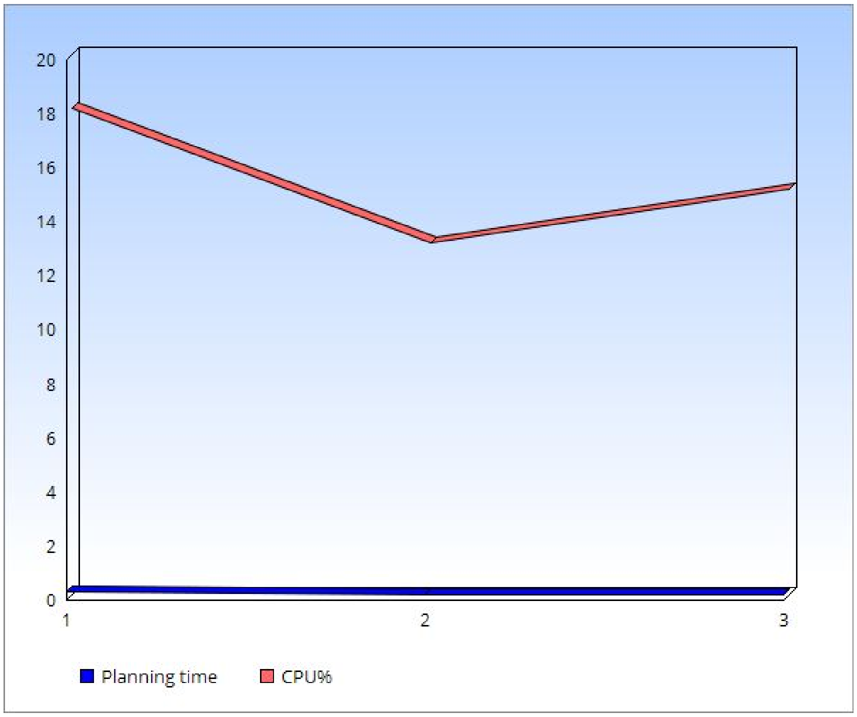
BiTRRT


24
22
20
18
16
14
12
10
8
6
4
2
0
1
2
3
Planning time
CPU%

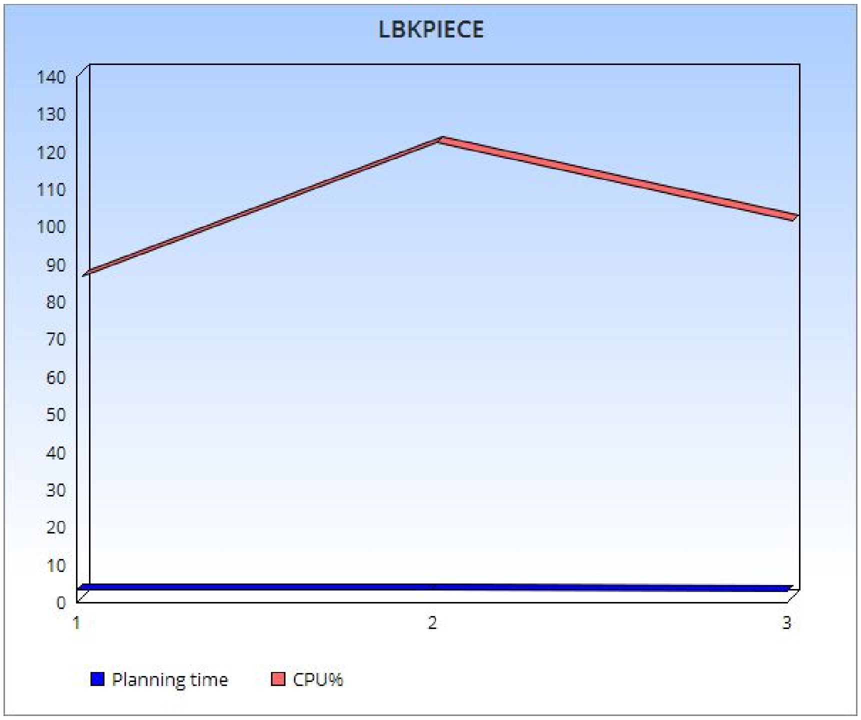
EST


320
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0
1
2
3
Planning time
CPU%

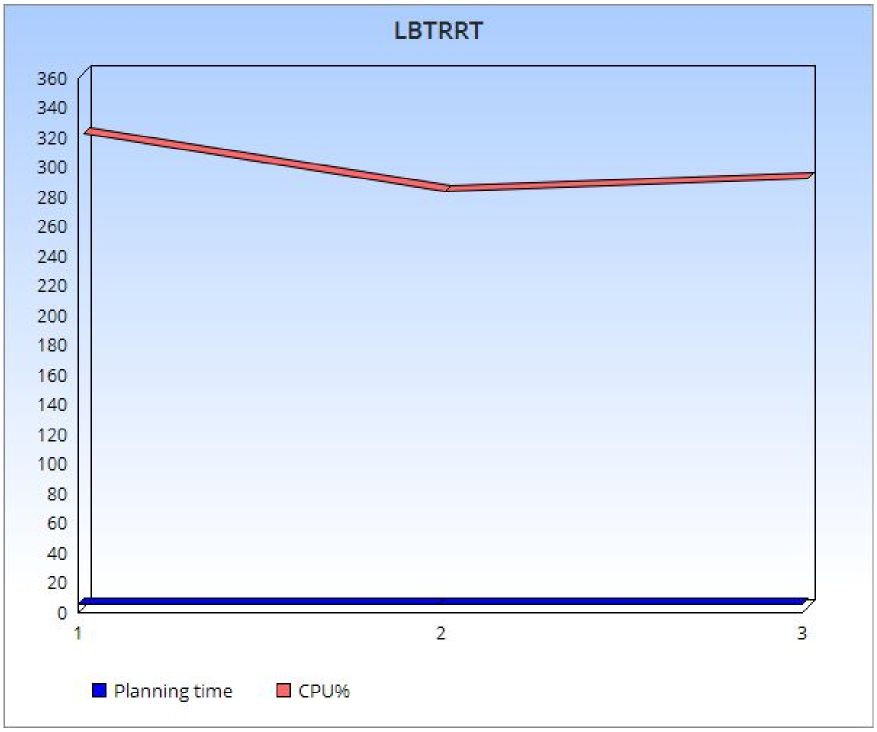
FMT


20
18
16
14
12
10
8
6
4
2
0
1
2
3
Planning time
CPU%

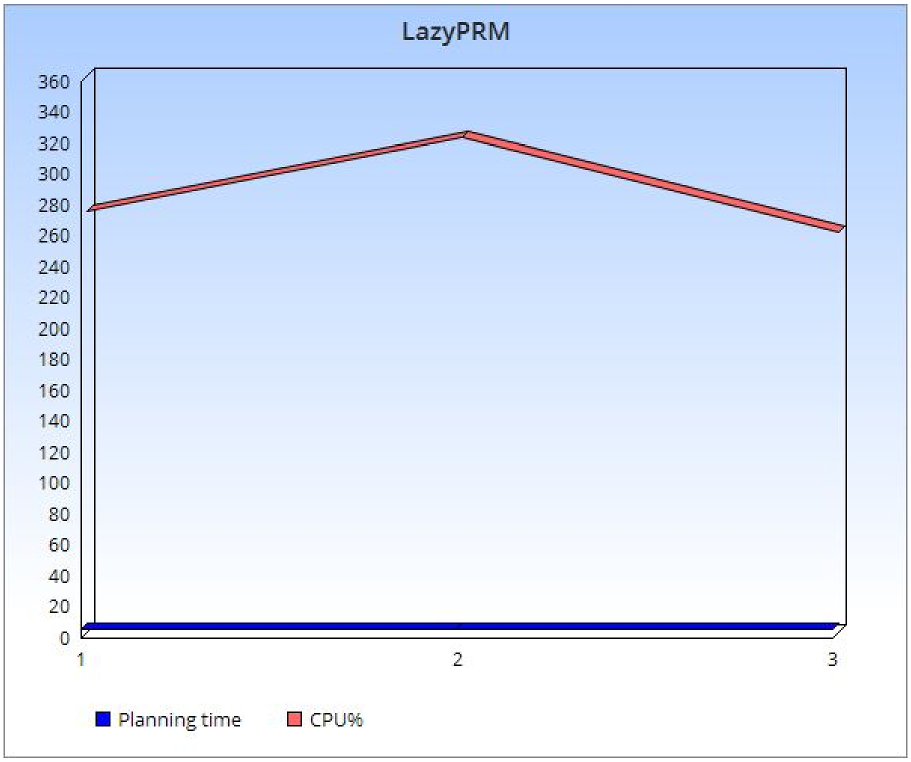
KPIECE



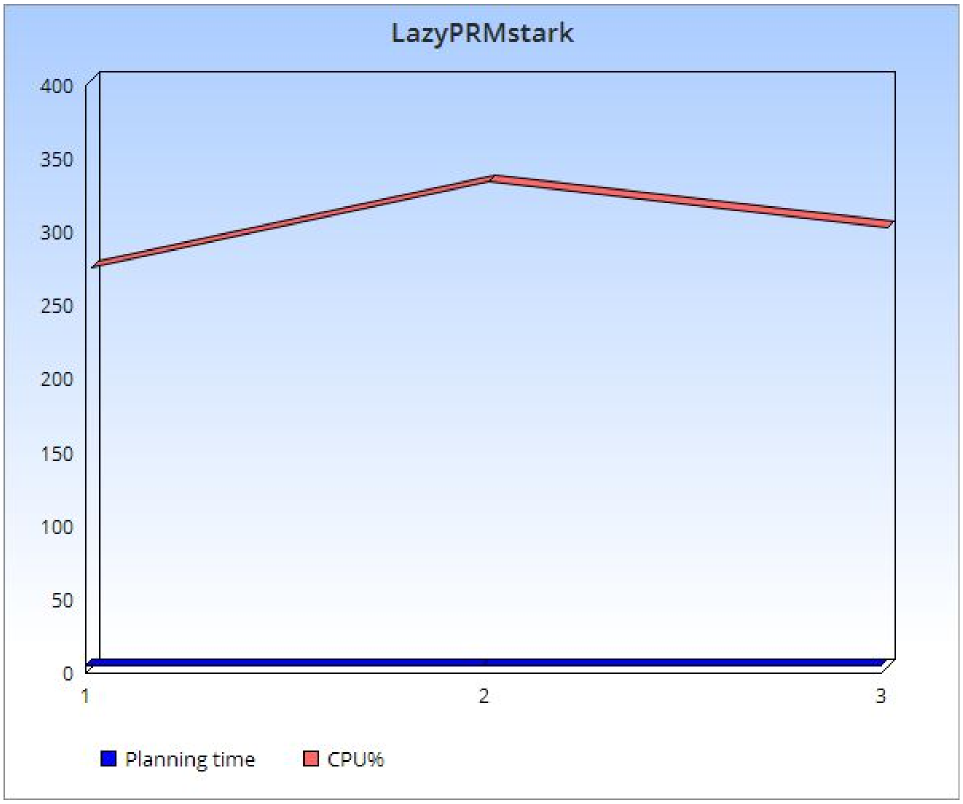
LBKPIECE

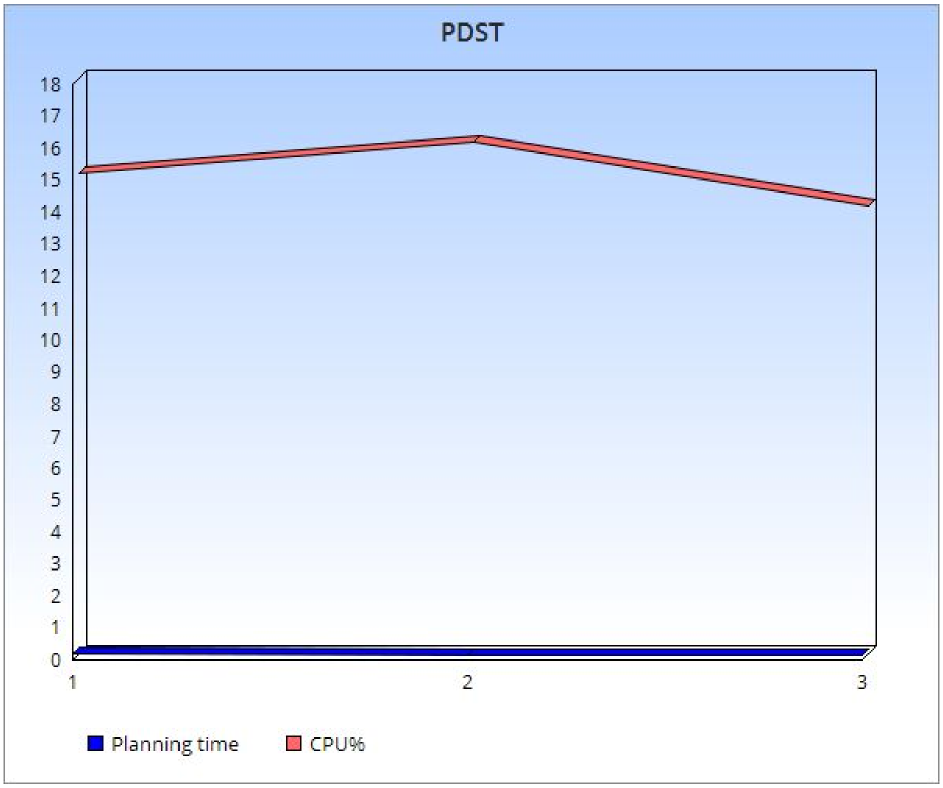


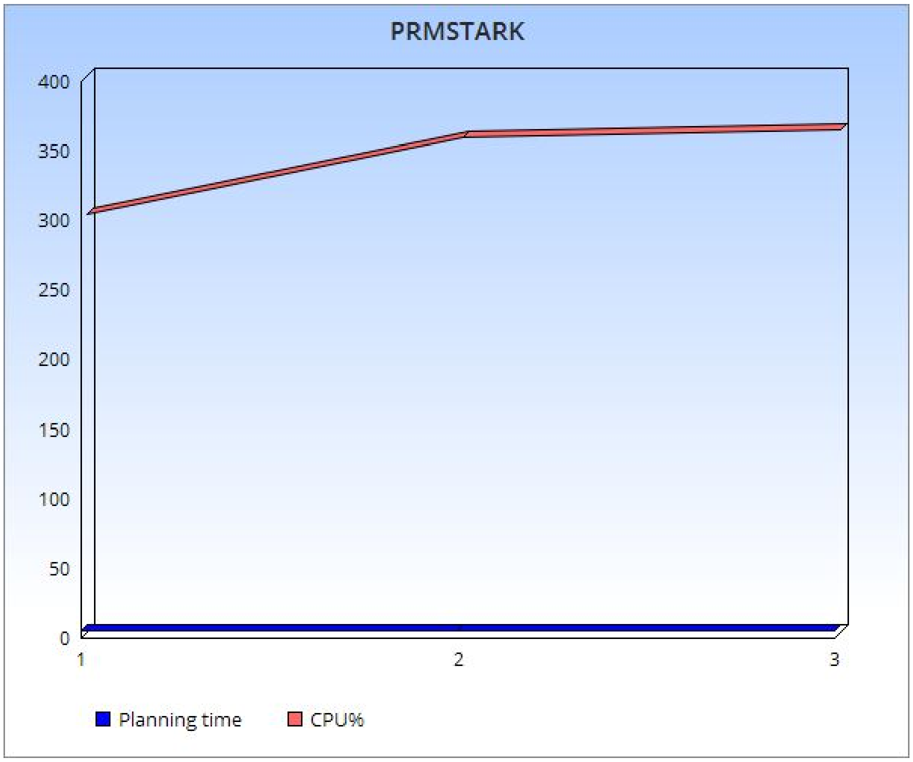
LBTRRT

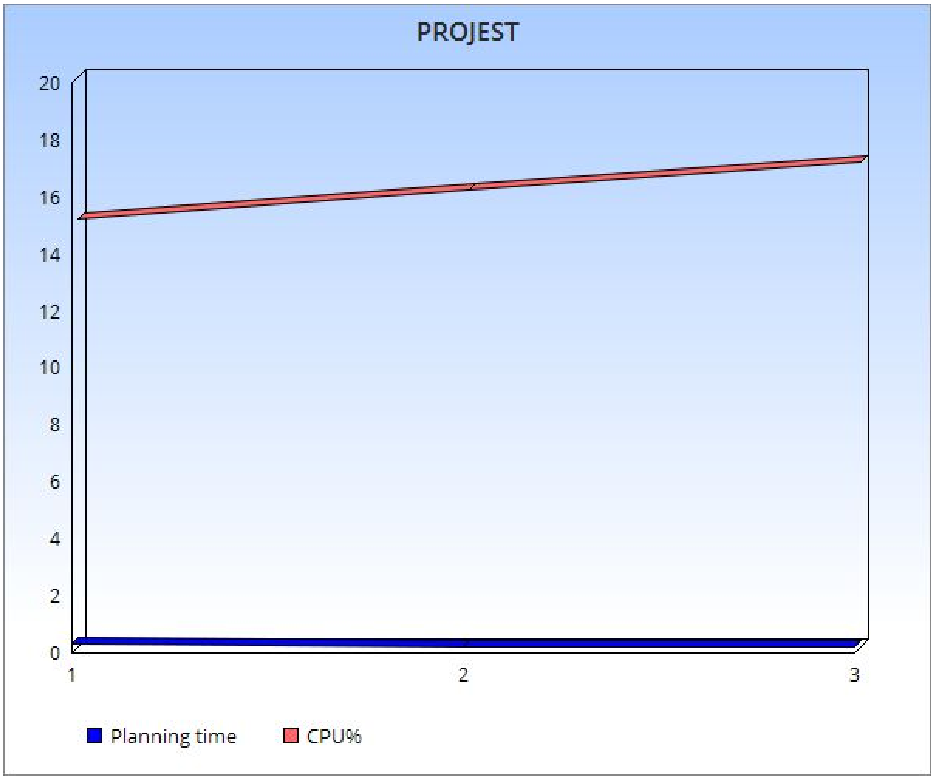

LazyPRM
360
340
320
300
280
260
240
220
200
180
160
140
120
100
80
60
40
20
0
1
2
3
Planning time
CPU%

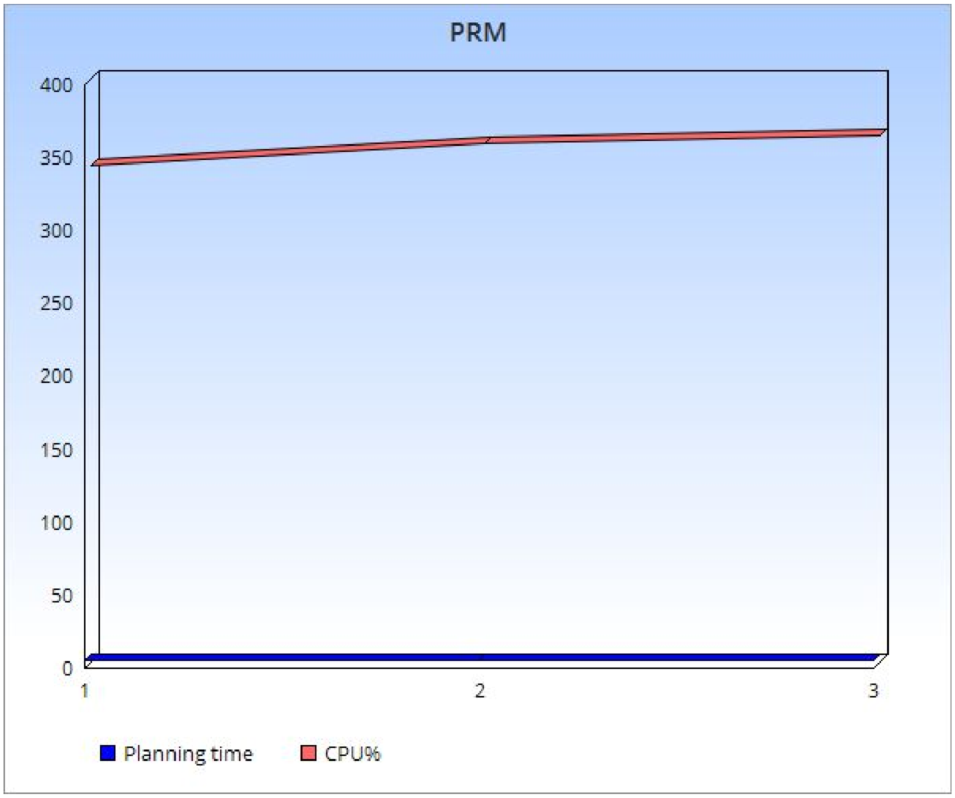
LazyPRM

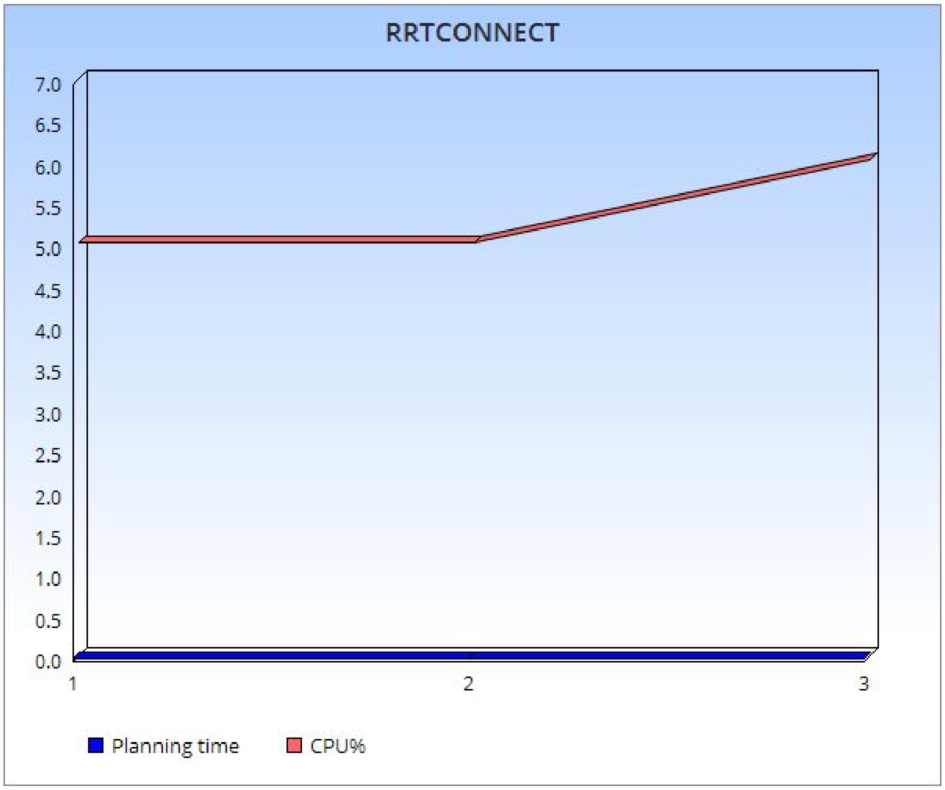

LazyPRMstark
400
350
300
250
200
150
100
50
0
1
2
3
Planning time
CPU%

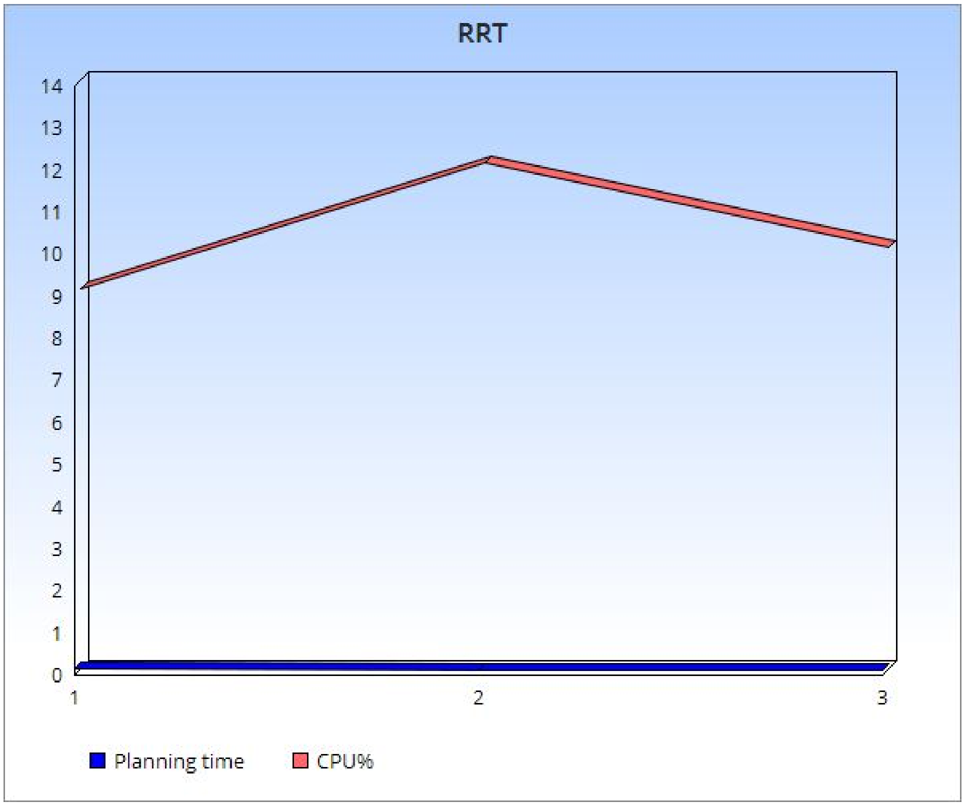

PDST
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
1
2
3
Planning time
CPU%

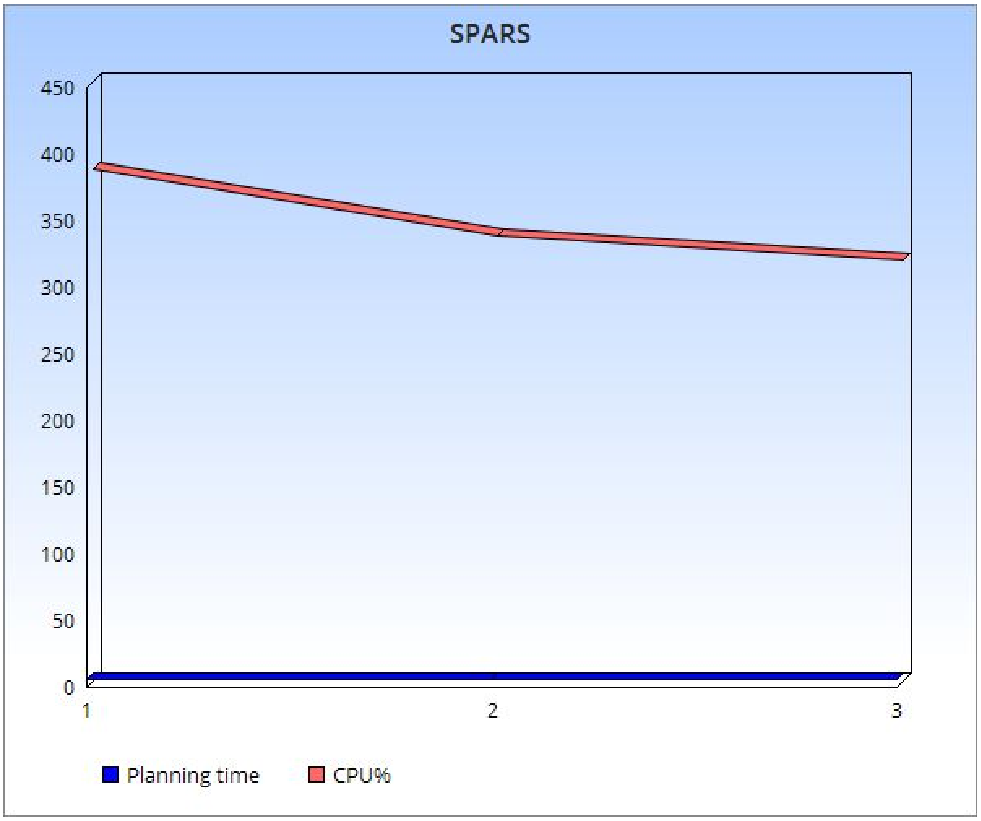

PRMSTARK
400
350
300
250
200
150
100
50
0
1
2
3
Planning time
CPU%

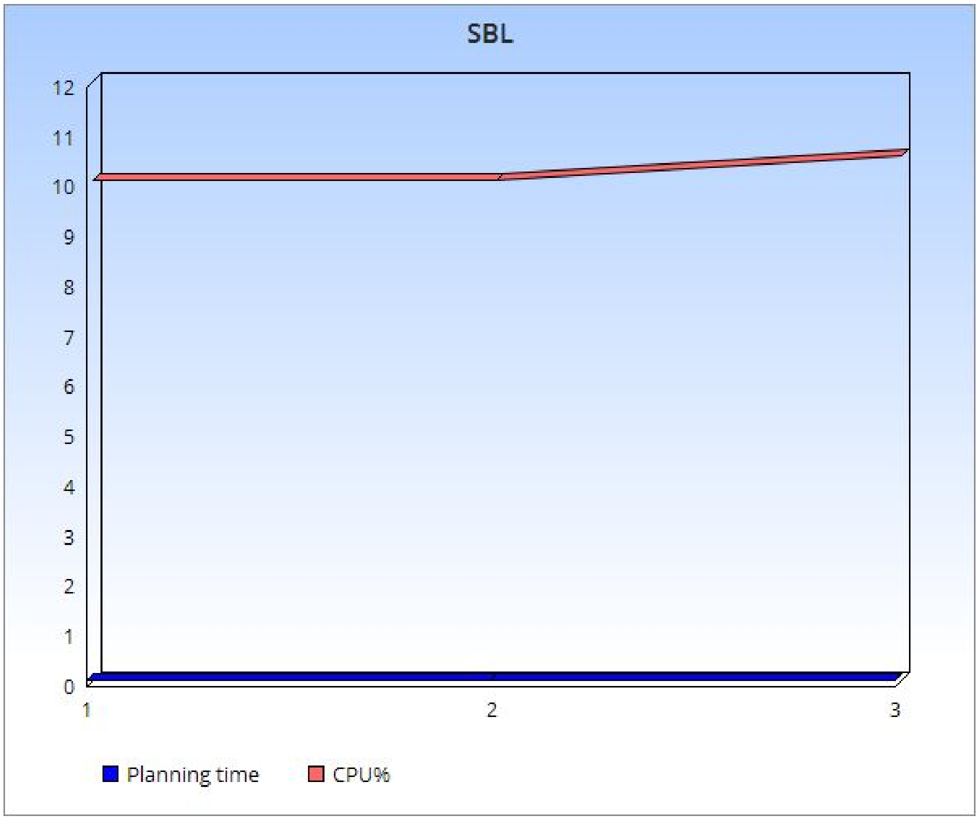

PROJEST
20
18
16
14
12
10
8
6
4
2
0
1
2
3
🟦 Planning time
🟥 CPU%

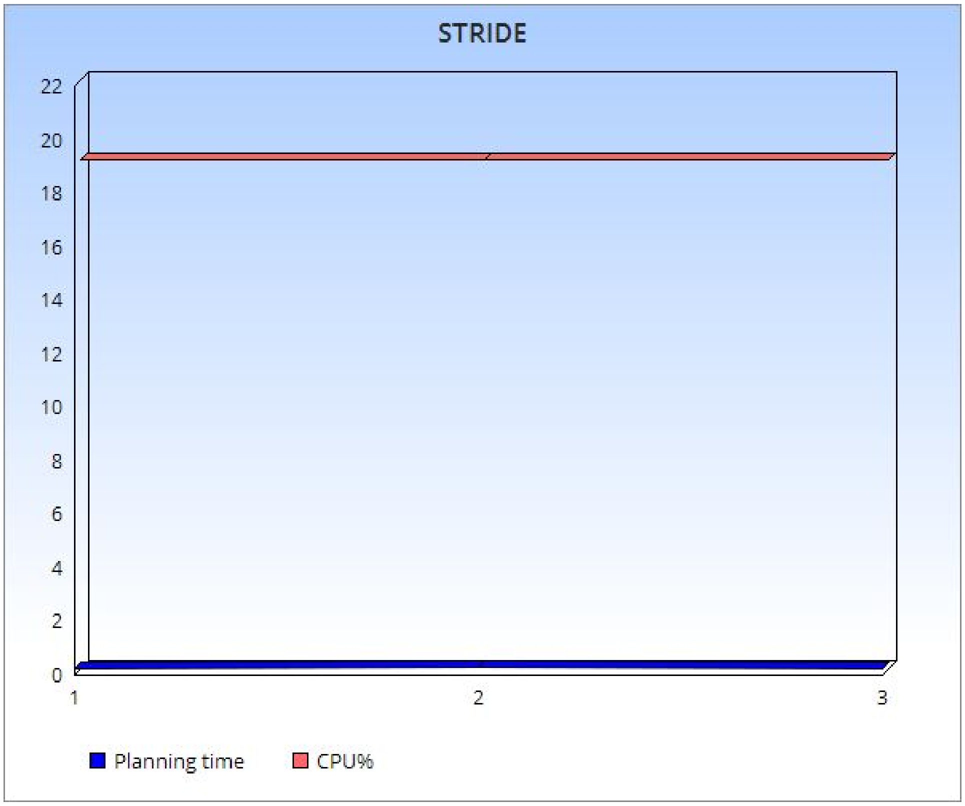

PRM
400
350
300
250
200
150
100
50
0
1
2
3
🟦 Planning time
🟥 CPU%

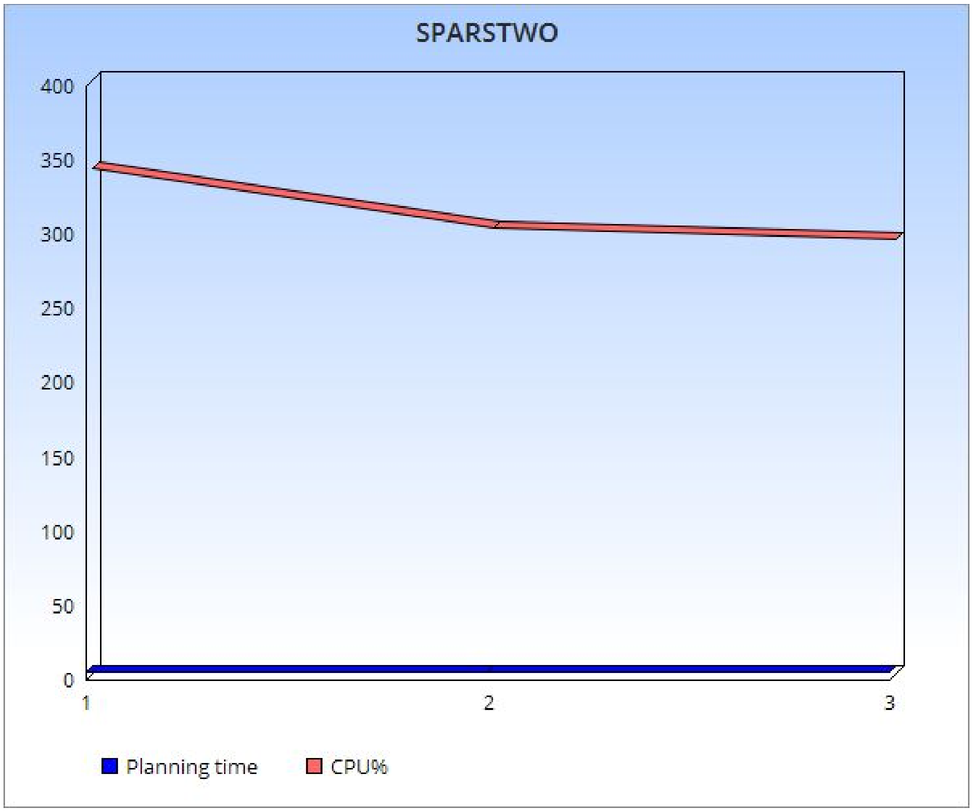

RRTCONNECT
7.0
6.5
6.0
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
1
2
3
Planning time
CPU%

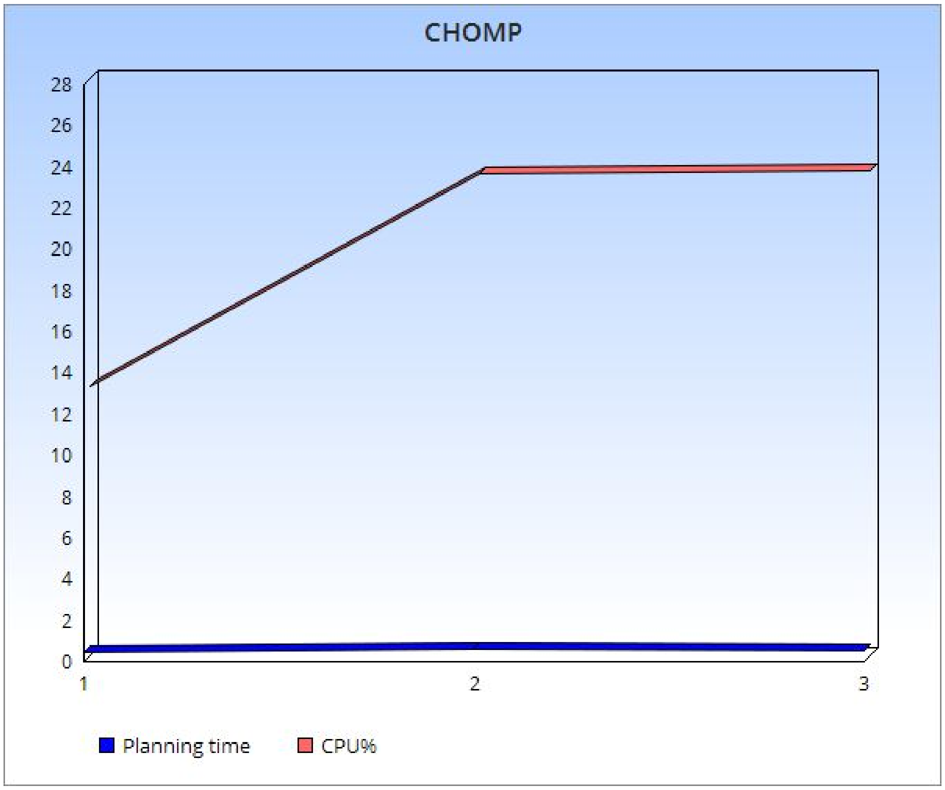

RRT
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
1
2
3
Planning time
CPU%

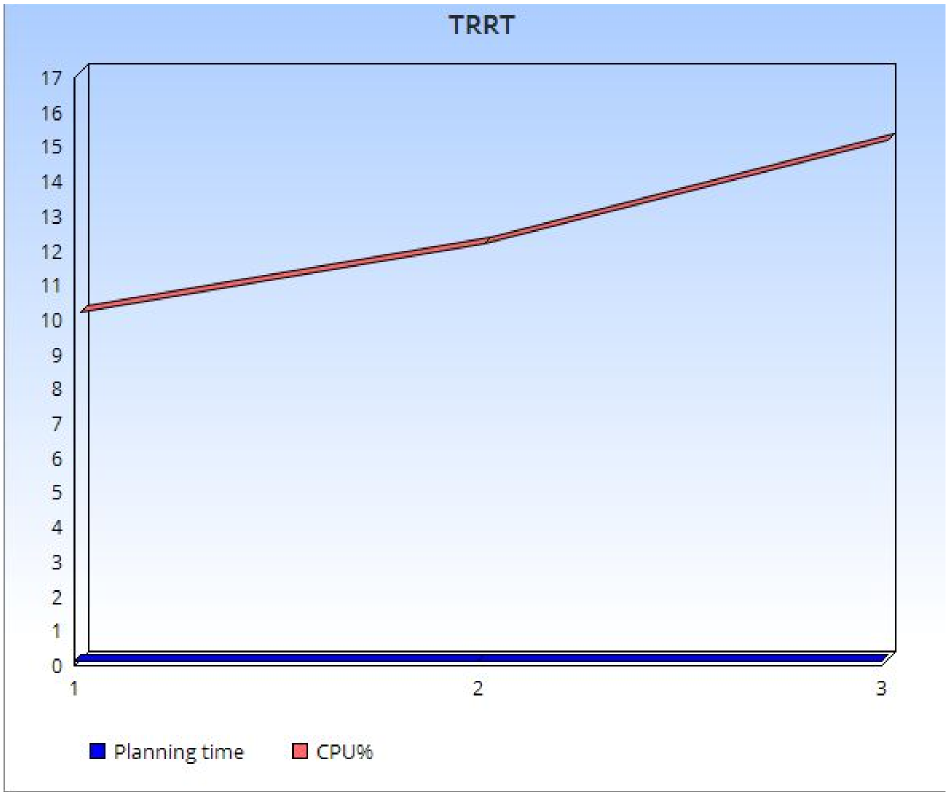

SPARS
450
400
350
300
250
200
150
100
50
0
1
2
3
Planning time
CPU%


SBL
12
11
10
9
8
7
6
5
4
3
2
1
0
1
2
3
Planning time
CPU%


STRIDE
22
20
18
16
14
12
10
8
6
4
2
0
1
2
3
Planning time
CPU%


SPARSTWO
400
350
300
250
200
150
100
50
0
1
2
3
Planning time
CPU%


CHOMP
28
26
24
22
20
18
16
14
12
10
8
6
4
2
0
1
2
3
Planning time
CPU%


TRRT
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0
1
2
3
Planning time
CPU%

Results :

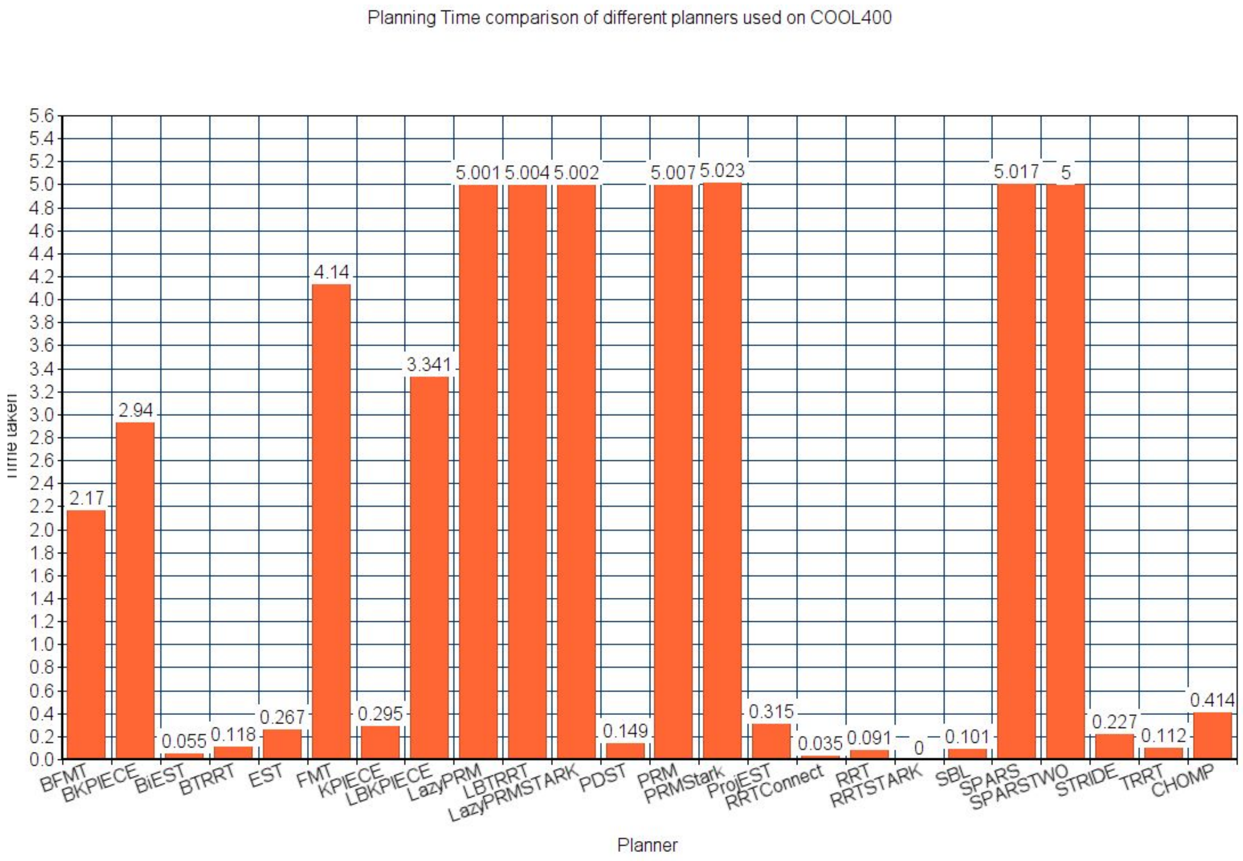


Fig 1.1 : Comparison of different planners on the basis of time taken to complete planning.

Different planners use different approaches as shown in [1]. In case of multi-query planners all the multi-query planners are able to complete the planning task in the same time. However, the CPU usage varies a lot as shown in *Fig 1.2*, the *LazyPRMStark* has the minimum CPU usage. It may be noted that planning fails in case of *RRTStark* planner for all the iterations. In case of Single-query planners *RRTConnec*t is the fastest according to the time taken to perform planning.

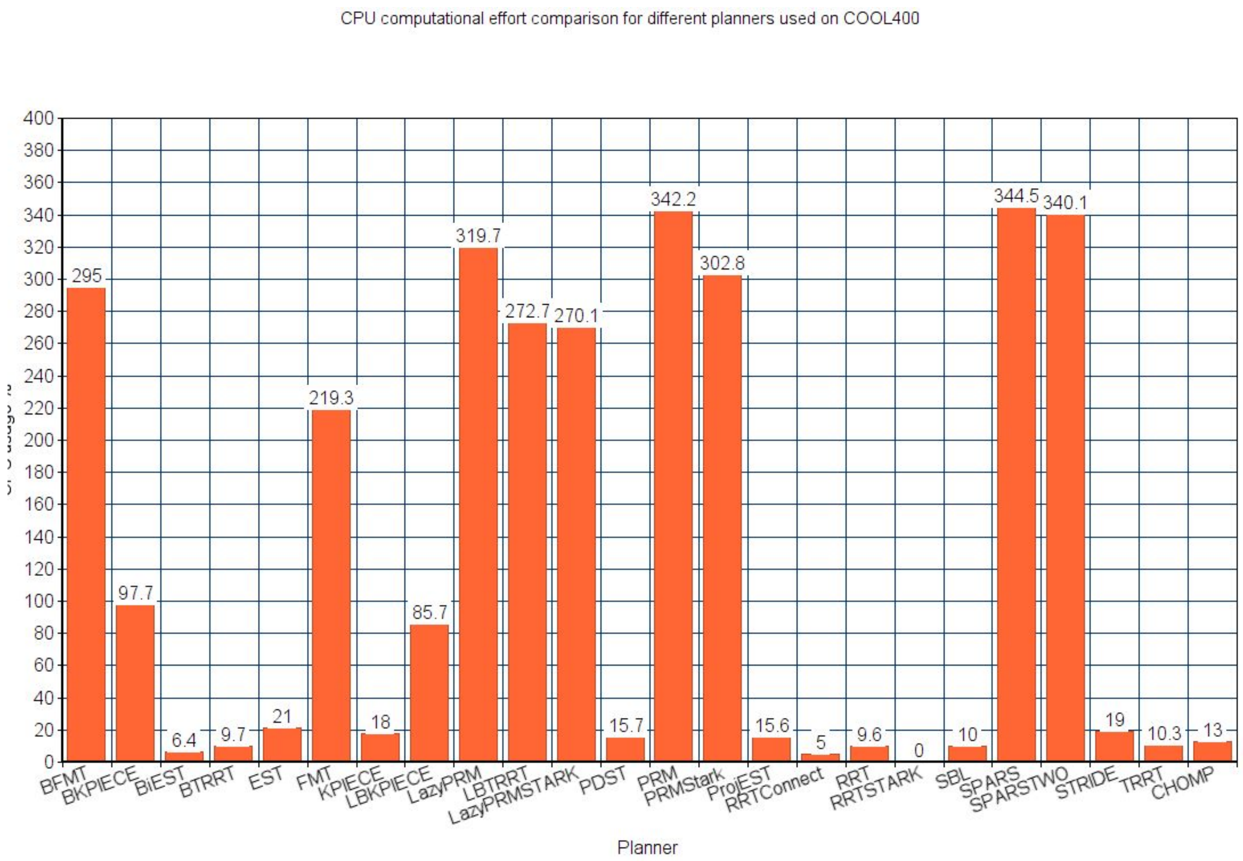
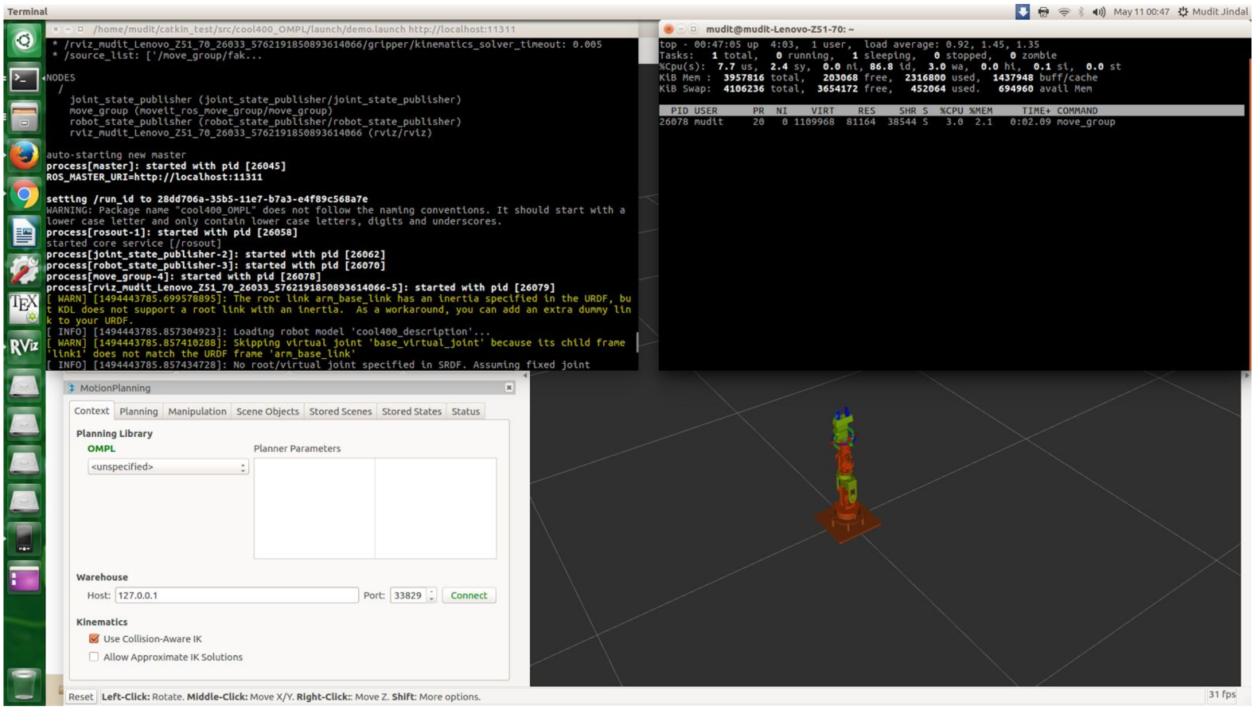


Fig 1.2: Comparison of different planners on the basis of CPU usage (%) of the move group node.

On the basis of the results shown in *Fig 1.2 RRTConnec*t uses the minimum CPU for its operation. Taking both computational efforts and planning time as parameters, it can be concluded that *RRTConnect* is the most efficient planner amongst single-query planners available in OMPL currently .This planner is a bidirectional version of RRT (i.e., it grows two trees).



```
 * /rviz_mudit_Lenovo_Z51_70_26033_5762191850893614066/gripper/kinematics_solver_timeout: 0.005
 * /source_list: ['/move_group/fak...

NODES
  /
    joint_state_publisher (joint_state_publisher/joint_state_publisher)
    move_group (moveit_ros_move_group/move_group)
    robot_state_publisher (robot_state_publisher/robot_state_publisher)
    rviz_mudit_Lenovo_Z51_70_26033_5762191850893614066 (rviz/rviz)

auto-starting new master
process[master]: started with pid [26045]
ROS_MASTER_URI=http://localhost:11311

setting /run_id to 28dd706a-35b5-11e7-b7a3-e4f89c568a7e
WARNING: Package name "cool400_OMPL" does not follow the naming conventions. It should start with a
lower case letter and only contain lower case letters, digits and underscores.
process[rosout-1]: started with pid [26058]
started core service [/rosout]
process[joint_state_publisher-2]: started with pid [26062]
process[robot_state_publisher-3]: started with pid [26070]
process[move_group-4]: started with pid [26078]
process[rviz_mudit_Lenovo_Z51_70_26033_5762191850893614066-5]: started with pid [26079]
[ WARN] [1494443785.699578895]: The root link arm_base_link has an inertia specified in the URDF, bu
t KDL does not support a root link with an inertia.  As a workaround, you can add an extra dummy lin
k to your URDF.
[ INFO] [1494443785.857304923]: Loading robot model 'cool400_description'...
[ WARN] [1494443785.857410288]: Skipping virtual joint 'base_virtual_joint' because its child frame
'link1' does not match the URDF frame 'arm_base_link'
[ INFO] [1494443785.857434728]: No root/virtual joint specified in SRDF. Assuming fixed joint
```

```
top - 00:47:05 up  4:03,  1 user,  load average: 0.92, 1.45, 1.35
Tasks:   1 total,   0 running,   1 sleeping,   0 stopped,   0 zombie
%Cpu(s):  7.7 us,  2.4 sy,  0.0 ni, 86.8 id,  3.0 wa,  0.0 hi,  0.1 si,  0.0 st
KiB Mem :  3957816 total,   203068 free,  2316800 used,  1437948 buff/cache
KiB Swap:  4106236 total,  3654172 free,   452064 used.   694960 avail Mem

  PID USER      PR  NI    VIRT    RES    SHR S  %CPU %MEM     TIME+ COMMAND
26078 mudit     20   0 1109968  81164  38544 S   3.0  2.1   0:02.09 move_group
```

Planning Library

OMPL

<unspecified>

Planner Parameters

Warehouse

Host: 127.0.0.1 Port: 33829

Kinematics

☑ Use Collision-Aware IK

□ Allow Approximate IK Solutions

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click:**: Move Z. **Shift:** More options.

Fig 1.3 Studying different planners and analysing CPU usage.

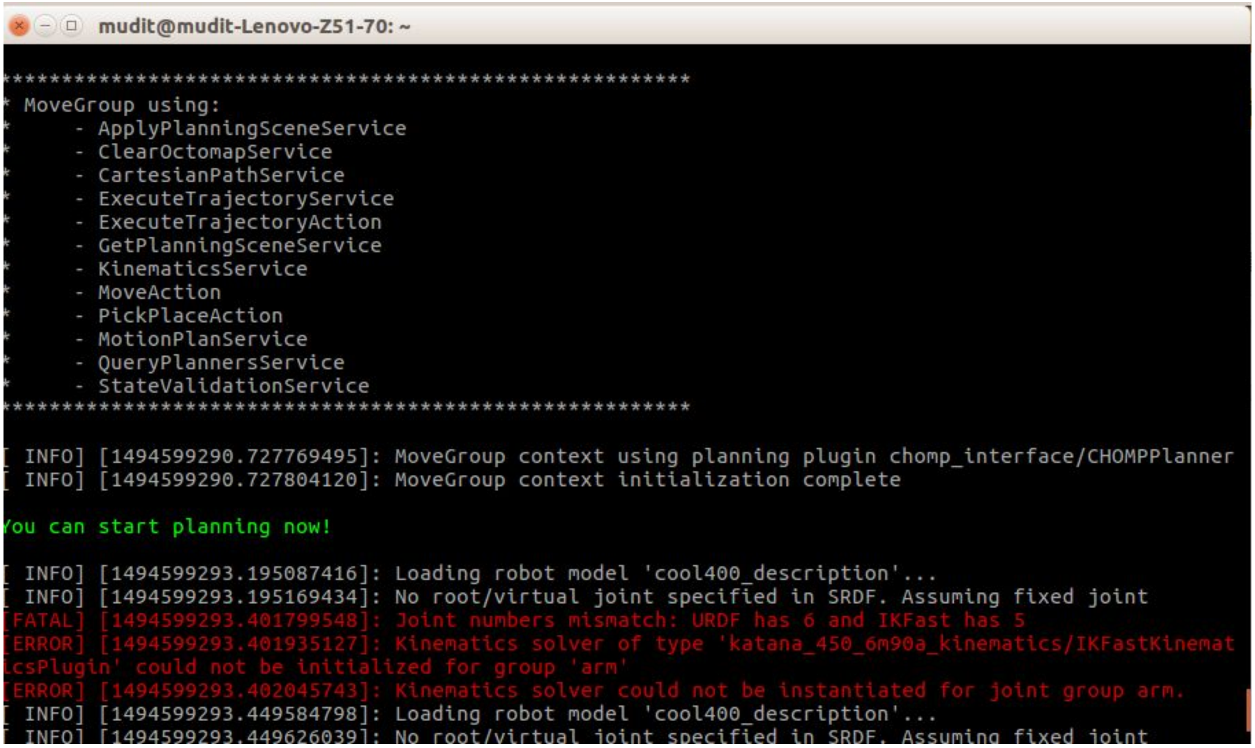
CHOMP is available in moveit but is still unstable. According to my research on the planner on COOL 400, the planner works fine and is stable on the arm. The path planned by the CHOMP is more smooth and natural looking as compared to OMPL planners. It provides a smooth motion to the arm.

Analysis of Kinematic solver :-

Tests and analysis was done on the following kinematic solvers:

1. Ik_fast
2. Trac_ik
3. Kdl
4. LMA kinematics

IK fast was not analysed due to “Joint numbers mismatch: URDF has 6 and IKFast has 5” as shown in the figure below:



```
*********************************************************
* MoveGroup using:
*     - ApplyPlanningSceneService
*     - ClearOctomapService
*     - CartesianPathService
*     - ExecuteTrajectoryService
*     - ExecuteTrajectoryAction
*     - GetPlanningSceneService
*     - KinematicsService
*     - MoveAction
*     - PickPlaceAction
*     - MotionPlanService
*     - QueryPlannersService
*     - StateValidationService
*********************************************************

[ INFO] [1494599290.727769495]: MoveGroup context using planning plugin chomp_interface/CHOMPPlanner
[ INFO] [1494599290.727804120]: MoveGroup context initialization complete

You can start planning now!

[ INFO] [1494599293.195087416]: Loading robot model 'cool400_description'...
[ INFO] [1494599293.195169434]: No root/virtual joint specified in SRDF. Assuming fixed joint
[FATAL] [1494599293.401799548]: Joint numbers mismatch: URDF has 6 and IKFast has 5
[ERROR] [1494599293.401935127]: Kinematics solver of type 'katana_450_6m90a_kinematics/IKFastKinemat
icsPlugin' could not be initialized for group 'arm'
[ERROR] [1494599293.402045743]: Kinematics solver could not be instantiated for joint group arm.
[ INFO] [1494599293.449584798]: Loading robot model 'cool400_description'...
[ INFO] [1494599293.449626039]: No root/virtual joint specified in SRDF. Assuming fixed joint
```

Fig 1.3 : IK_fast error

TRAC is a faster and a better version of KDL and therefore outperforms KDL in many aspects. KDL is not able to solve for some of the states when gripper was moved manually in Rviz. The performance of LMA and TRAC ik is comparatively same. However, TRAC-IK slightly outperforms LMA in terms of the time taken to solve kinematics. Therefore, TRAC IK was chosen as kinematic solver in moveit.

References

[1] : Available planners in OMPL - http://ompl.kavrakilab.org/planners.html