4th IFAC Conference on Modelling and Control in Agriculture,
Horticulture and Post Harvest Industry
August 27-30, 2013. Espoo, Finland

Task and Motion Planning for Apple Harvesting Robot ⋆

Tien Thanh Nguyen, Erdal Kayacan, Josse De Baedemaeker
Wouter Saeys

MeBioS (Mechatronics, Biostatistics and Sensors) Division,
BIOSYST (Biosystems) Department, KU Leuven, Belgium
(Tel: +32 163 21444; e-mail: tienthanh.nguyen@biw.kuleuven.be)

Abstract: Selective harvesting for seasonal fruits like apples requires intensive manual labour in a short period; however, due to the delicate property of the fruits, it cannot be performed with an existing commercial machine, which urged to be replaced by robots. The goal of this study is to introduce a framework for motion and hierarchical task planning which allows the manipulator to pick apples in the orchard. The hierarchical task planning assures that the manipulator performs the harvesting task in the higher control level in corporation with other components: sensors system, mobile platform while dealing with the complicated and uncertain environment. The motion planning provides the abilities to the manipulator: to avoid the obstacles, to reach the targets, and to perform the detaching movement elaborated from a limited number of pre-defined strategies. This motion and task planning framework has been successfully evaluated in simulation, and the real-time tests show that the harvesting task is accomplished with assured communication between sensing, planning and executing.

Keywords: motion planning, robotics, apple, harvesting

1. INTRODUCTION

Automation of harvesting seasonal fruits, e.g. apples is highly desirable for European countries due to the difficulties of finding workforce for manual harvesting in a short period and the increase in the farm labour cost. In order to be able to automate the harvesting process, harvesters with mass harvesting methods, i.e. rotating beater bar or trunk shaker, have been developed. However, these methods only suit for cider apples. Considering the high quality fresh apples, bruising in the fruit is not allowed. Therefore, a selective harvesting method (*apple by apple*) in which the machine can mimic the picking task of human is preferred. Thanks to the rapid developments in robotics, the automated robotic systems using a mobile platform carries one or several manipulator(s) is regarded as a potential replacement to release humans from the monotonous fruit harvesting task.

The research on harvesting robots started more than 20 years ago and has resulted in various prototypes. For example, many fruit harvesting robots for tomatoes, eggplants, lettuces, etc. were developed by Kondo and Ting (1998), for melons by Edan et al. (2000), for apples by Baeten et al (2007) and by Zhao et al. (2011). However, the developed prototypes still cannot meet the challenging requirements for practical applications, i.e. combination of high speed and accuracy with ability to deal with the complex, dynamic and continuous changing tasks and environment for an acceptable cost. Two major challenges that the harvesting robots have to solve are the implementation the domain knowledge of the harvesting task to be able to mimic the human picker and dealing with the uncertainties and occlusions in the environment. Nguyen et al. (2012) presents that the domain knowledge of the harvesting task is lied in grasping and detaching movement, and the success of the implementation is determined by the design of special end-effector and pre-defined detaching movement. However, the apples-harvesting robot works in the outdoor environment resulting in a fact that it is impossible to sense the environment sufficiently to cope with all of the critical uncertainties. Thus, the robot must actively update the information while executing actions, and continuously use updated information to control or make decision about the action. To support that concept, a framework that integrates the task planning in the logical space with the motion planning in the geometry space is required.

Advances in perception, navigation and motion planning have enabled sophisticated robotic systems that interleave perception, planning and acting to perform required task in realistic domains (Conor et al (2009), Srinivasa et al. (2010)). Nevertheless, most of the manipulation systems mentioned above work only in indoor environment with rather simple pick-and-place task. Some recent researches in robotics, i.e. Kaelbling and Lozano-Perez (2011) about integration robot task and motion planning by using hierarchical planning and continuously geometric operating, Xue et al. (2012) about integration motion and grasp planning, are promising for solving the challenges of the

⋆ This research is funded by European Commission in the 7^{th} Framework Programme (CROPS GA no 246252).
The manipulator used in this paper is developed by the Institute of Applied Mechanics at Technical University of Munich.

harvesting robots, however there is still a gap to implement them in practical applications.

The goal of this paper is to introduce a framework for motion and task planning applied for the apple harvesting robot. The hierarchical task planning assures that the manipulator performs the harvesting task in the higher control level in corporation with other components: sensors system, mobile platform while dealing with the complicated and uncertain environment. The proposed task planning framework allows the system to be modular, whereas the task can be easily modified with respect to the requirements of the operation, and is defined independent of the hardware. Besides, the motion planning provides the abilities to the manipulator: to avoid the obstacles, to reach the targets, and to perform the detaching movement elaborated from a limited number of pre-defined strategies. The proposed motion planning framework uses the kinematic model of the robot and the collision map generated from the sensor signals to generate a collision-free joint trajectory that can be implemented into the low level controller of the manipulator. This motion and task planning framework has been successfully evaluated in simulation, and the real-time tests show that the harvesting task is accomplished with assured communication between sensing, planning and executing.

The paper is organised as follows: In Section 2, the hardware components and the control system of the robot are represented in detail. In Section 3, the framework for the hierarchical task planning; and the motion planning for the robot is described. In section 4, the implementations and results of the framework are discussed. Finally, in Section 5, conclusions and suggestion for the future research are made.

2. THE APPLE HARVESTING ROBOT

The conceptual design of the apple harvesting robot has been defined within the Crops project [1] in order to have a highly configurable, modular robotic systems that are capable of adapting to several agricultural tasks and conditions. In this design, a mobile platform equipped with power and compressed air supplies that can easily travel in the apple orchard, carries a custom built manipulator with the sensors, vision and control system. For the demonstration, the robotic harvesting system will first be tested as a '*stop and go*' system to reduce the complexity in sensing and manipulation.

2.1 The manipulator

In this study, a redundant modular multi-purpose agricultural harvesting manipulator with 9 degrees of freedom (DOF) developed by the Institute of Applied Mechanics at the Technical University of Munich (Baur et al. (2012)) is used as shown in Fig. 1. The working area of an apple harvesting robot covers whole distribution space of ripe fruits on the tree. The redundant DOFs allow the manipulator to reach the fruit from various directions, and be able to operate at close range as well as at a distant location with a high dexterity which is very important to perform the detaching movement for apples picking. The first joint q_1 is used for uplifting the whole manipulator, in order to reach the highest fruits on the tree which are 3m height in our testing orchard. The second joint q_2 is used to move the manipulator to both sides respect to the picking direction which helps to increase the operation space. On the other hand, the joint $q_3 - q_5$ move the manipulator forward and backward in the picking direction which allow to reach deeper inside canopy and also to avoid the obstacles near the manipulator. The joint $q_6 - q_8$ enable high dexterity in close range with the target. The joint q_9 rotates the end-effector to the proper grasping orientation to perform the detaching movement.

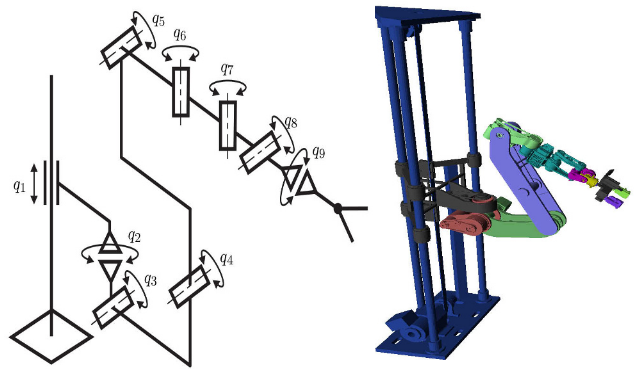

Fig. 1. Kinematic diagram and 3D model of the apple harvesting manipulator

2.2 The control system

The structure of the control system of the robot is shown in Fig. 2 which consists of two layers: planning layer and execution layer. At the centre of execution layer, a real-time control unit calculates the inverse kinematics of the manipulator and communicates with the low-levels of each joints by four CAN bus. The main unit of the planning layer is a host computer using ROS [2] as the environment to integrate the control interface and all of software

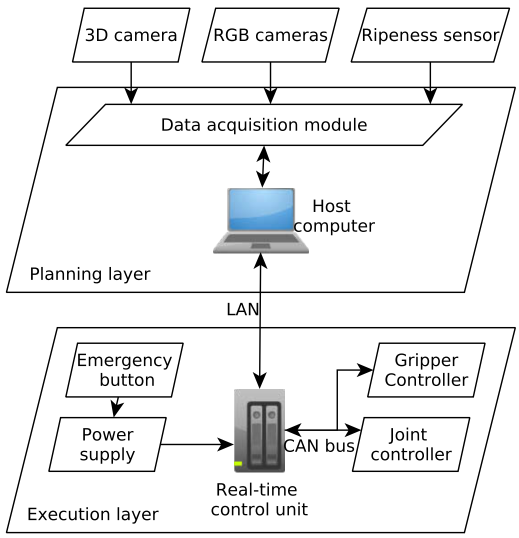

Fig. 2. Structure of the control system

[1] http://www.crops-robots.eu/

[2] http://www.ros.org/

modules. The sensor systems includes 3D cameras, RGB cameras, ripeness sensors, etc. used to collect environment information. The real-time control unit connects to the host computer through a LAN interface that provides the status information of the manipulator and receives the desired pose of the Tool Center Point (TCP) for inverse kinematic calculation. However, the host computer can also send the desired states of each joint of the manipulator (positions, velocities, accelerations) to the execution layer without using the inverse kinematic calculation module of the real-time control unit. This option allows the control of the state of the manipulator by the feedbacks coming from the environment i.e. visual servoing.

2.3 Control flowchart

The control flowchart of the apples harvesting robot defines the sequences of operations that the robot must follow to process the harvesting task. Figure 3 presents our simplified flowchart for apple harvesting robot. The implementation of this flowchart using the hierarchical task planning are described in Section 3.1.

In order to harvest all the detected fruits, the control flowchart has been constructed to allow the following issues:

- Fruits can be detected with both fixed cameras and cameras mounted on the gripper that permits to detect the fruit hidden inside the tree canopy and can not be seen by the fixed cameras.
- The properties of the detected fruits such as: position, orientation, size, etc. are stored in a target list data which can be updated and reorganized. By handling the target list data, it is assured that the robot will not leave any detected fruit on the tree and has data to perform the high level planning in parallel with executing current task.

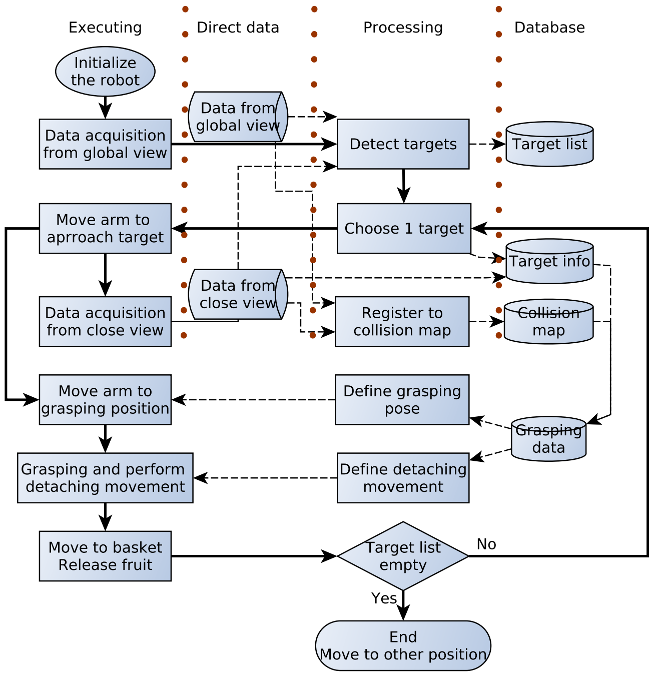

Fig. 3. Simplified flowchart of apple harvesting robot

- Obstacles are represented by 3D collision map which also can be updated while more data are acquired from cameras.

All of these issues belong to the perception module of the robot which is also defined as the bottleneck of any harvesting robot system.

3. FRAMEWORK FOR TASK AND MOTION PLANNING

3.1 Hierarchical task planning

A hierarchical approach is proposed to solve the challenges in implementation domain knowledge into robotic task and environment with uncertainties and occlusions. By performing a temporal decomposition at multiple levels of abstraction, domain knowledge are always addressed in reasonable subtasks which help the planning and implementation feasible. Moreover, simple subtask allows to provide the planning with execution. If any unexpected effect appears, a new plan can be reconstructed and the execution resumes. Figure 4 shows the hierarchical planning and execution tree for apple harvesting robot. In this figure, each task/subtask could be considered as a state machine in which the meaning of state depends on the level of abstraction. Each state machine includes the information as follows:

- **activate bit**: *ON* when the state machine is set active, *OFF* when the state machine is deactivated.
- **precondition**: list of conditions under which the state machine is set active
- **end-condition**: list of conditions under which the state machine is deactivated
- **input**: input data that the state machine acquires from the components or subsystems
- **effect**: effects of the state machine to data stream

The **input** and **effect** of the higher level task include the **input** and **effect** of its subtasks. The **activate bit** of a higher level task is ON when at least one of the **activate bit** of its subtasks is ON. There are 4 tasks in Fig. 4 presented in orange colour. These tasks could be simplified as '*move the end-effector from current position to an another defined position*'. The motion control of the manipulator in these tasks is defined in the motion planning framework in Section 3.2. Two tasks presented in blue colour in Fig. 4 are used to control the gripper.

To achieve modularity and reusability, the harvesting task has been divided into four sub-domains:

- Target list: handles the target of the harvesting task, includes fruit detection (location, orientation, size, etc), detaching situation recognition. The detaching situation derived from sensing data of objects present in action radius will be used to decide which detaching strategies would be used.
- Collision map: the collision map includes the compressed 3D point cloud data with annotations of the current target, the future targets with pre-planning order and other obstacles.
- Navigation: handles the arm navigation in the collision map. This provides the manipulator ability to

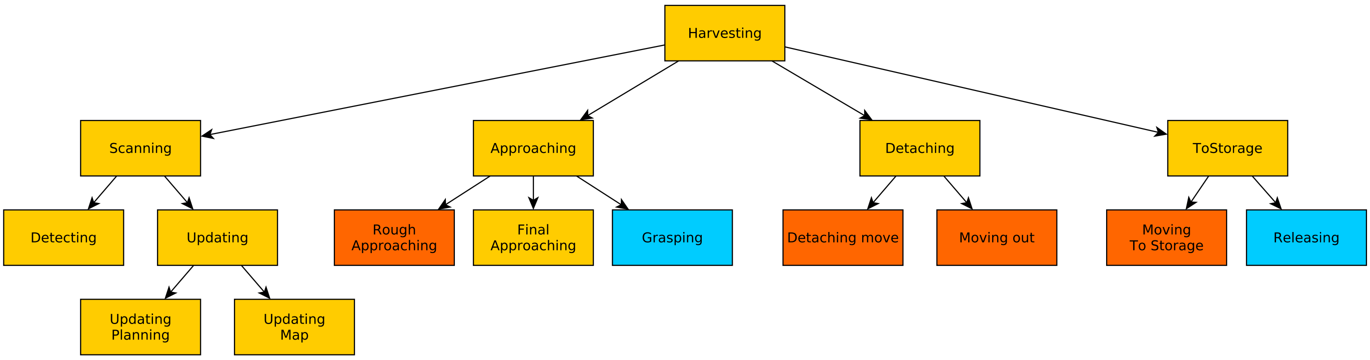

Fig. 4. Hierarchical planning and execution tree for apple harvesting robot

approach target, perform the detaching movement based on the pre-defined strategies.

- Task planner: handles the sequence and schedule of sub-tasks, chooses the predefined detaching movement and plans the order of the targets. It also includes the condition monitoring for each sub-task.

The sub domains mentioned above allow us to re-use and re-apply the high level control architecture into different robots with different tasks. In each component of the state machine, the perception, the control and the planning operations are involved which grants the reactivity of the overall system.

3.2 Motion planning

A motion planning framework has been built based on the open motion planning library - OMPL (Sucan et al. (2012)). The unified robot description format - URDF model of the manipulator, which defined the kinematics relations between different links and joints of the manipulator; information and positions of cameras, sensors, is required for the configuration. Number of the sampling-based algorithms such as rapidly-exploring random trees (RRT), bidirectional RRT (RRTConnect), kinematic planning by interior-exterior cell exploration (KPIECE), bidirectional KPIECE (BKPIECE), lazy bidirectional KPIECE (LBKPIECE), single-query bidirectional lazy collision checking planner (SBL) and expansive space trees (EST), etc. can be implemented inside the framework and can be easily changed depending on the requirements of the task. Self-collision checker and obstacles avoidance are included inside the framework. The inputs are the 3D CAD model of the manipulator and the collision map generated from the sensor data. To be used in motion planning for the manipulator, the collision map must satisfy these following requirements:

- **using probabilistic representation**: to deal with the uncertainties such as noises from measurements, dynamic obstacles and to integrate data from multiple sensors of the robot
- **modelling of free, occupied and unmapped area**: to plan the collision-free paths for the manipulator only in the area that have been detected to be free. The occupied and unmapped area should be avoided, thus, should be represented in separated types.
- **efficient in access and storage**: multi-resolutions representation of the collision map will help reduce time in access and collision checking, since navigation for the manipulator needs more details in area near target and manipulator than area far away.

In our implementation, we use OctoMap (Hornung et al. (2013)) - an probabilistic 3D mapping framework based on octrees to represent collision map. The point clouds, acquired from several sources and registered into global coordination, are fused to OctoMap mapping server, and will be used by the motion planner in the arm navigation and grasping pipeline.

The output of the motion planner is the trajectory of each joint of the manipulator. The sampling frequency is set to 0.05 seconds in order to send to the low level controller as the desired trajectory of each motor.

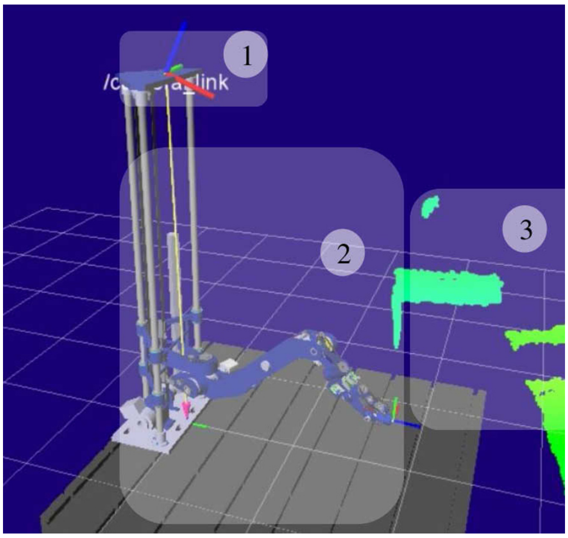

Fig. 5. A snapshot of the visualization of the manipulator 1: Camera link, 2: 3D model of the manipulator, 3: Point cloud data acquired from camera

The detaching movement is simplified into a motion planning problem: move the end effector from the grasping position to another position. This position is derived from the requirement of the contact between a "finger" of the gripper and the stalk of the apple to create the necessary shear force to break the stalk at the abscission zone (Nguyen et al. (2012)).

4. RESULTS

The results of these frameworks are a task planner for the apple harvesting robot with a concrete state machine of the system and a motion planner which can guide the manipulator to perform the related task.

Thanks to the modularity of the task planner, each subtask can be tested separately. A simulation study has been developed in Gazebo which is a multi-robot simulator for outdoor environment. It includes the kinematic and 3D simulation of the manipulator, apple orchard environment, and simulation of the sensor system: Kinect sensor, laser scanner and Swissranger SR4000 3D camera, 3D ToF camera on the manipulator. Figure 6 shows the robot in simulation with the acquired data from the RGB camera and ToF camera.

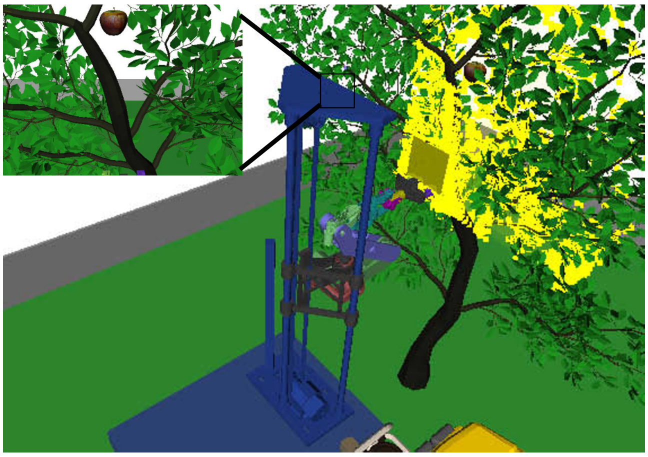

Fig. 6. Simulated robot and environment with acquired data from RGB and ToF cameras

The implementation results are encouraging. The testing situation is described in Fig. 7 in which the model of an apple tree is 1m away from the robot. In this figure, only the models of tree trunk and branches are considered as obstacles. A set of planning queries, which are the picking situations and move to bin, is created. Several sampling-based planners are implemented to search for the most efficient algorithm for testing situation: RRT, RRTConnect, KPIECE, BKPIECE, LBKPIECE, SBL and EST. For this testing, each planner has been tested a total of 100 planning queries (harvesting 50 apple fruits and move them to the collecting bin). All of these algorithms were run using the default parameters, the resulting trajectories were post-processed (simplifying and interpolating) by the same trajectory filter algorithm.

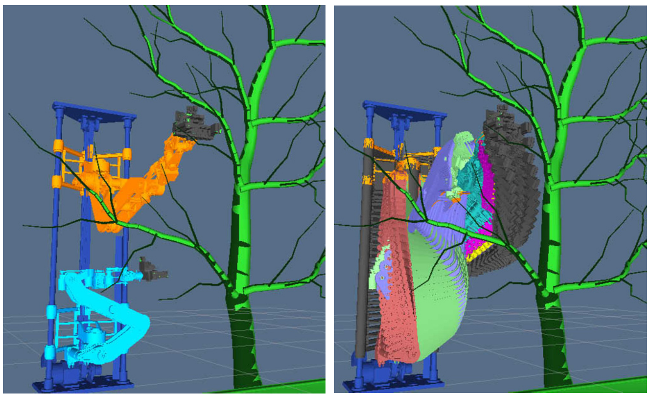

Fig. 7. Motion planning situation

The evaluation results are shown in Figs. 8 and 9 with 2 comparison factors: total runtime (including the planning time and the trajectory filtering time) and success rate with 5 seconds of the time limit for the planning time. Due to the fact that the testing situation is fairly easy with 1m of free space in front of the manipulator and the manipulator with 9 DoFs has sufficient flexibility, most of the planners have 100% of the success rate in 5 seconds. In this test situation, the RRTConnect planner is the most efficient algorithm with 100% of success rate and also the fastest algorithm with approximated total running time around 1 second with 0.25 seconds of approximated planning time.

The task planner successfully communicates with the real-sensor data and the manipulator. In Fig. 10, a demon-

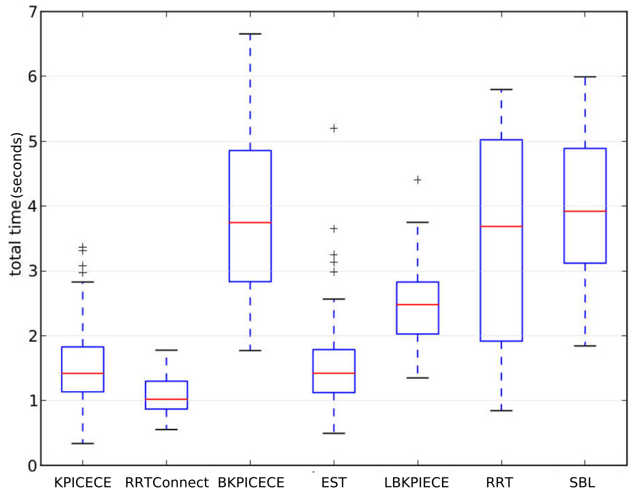

Fig. 8. Total running time of different planners

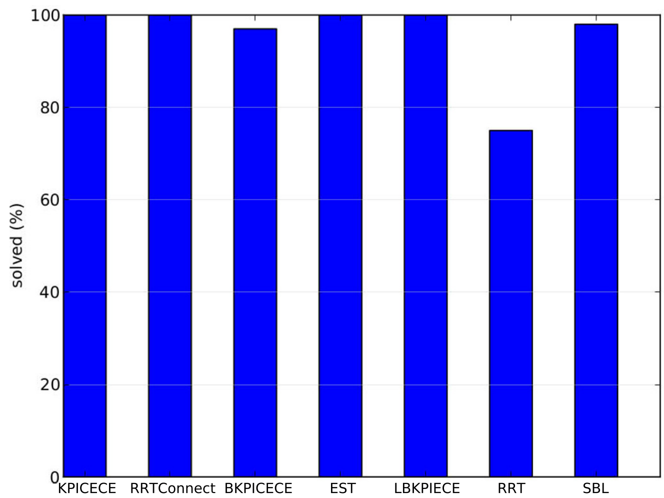

Fig. 9. Success rate of different planners

Fig. 10. Testing manipulator in apple orchard

stration of the manipulator approaching the apple fruit is shown. The joint trajectory output of the motion planning can be received and executed in the low level controller of the manipulator. Figure 11 represents the planned and measured positions of the joints from 2 to 7 of the real manipulator.

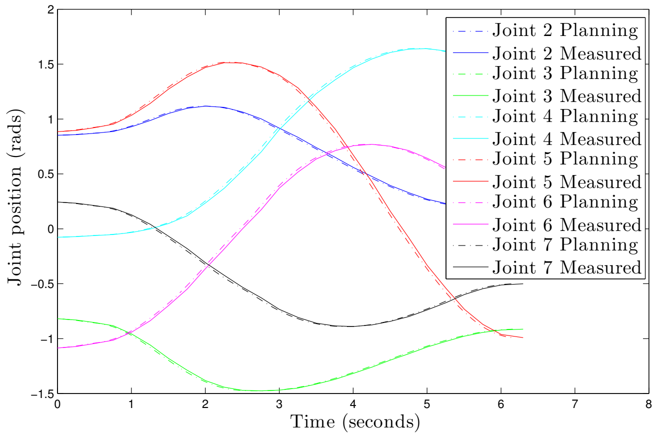

Fig. 11. Implementation of the motion planning in the real-time system: Planning and measured joint position

5. CONCLUSIONS AND FUTURE WORKS

In this study, a framework for motion and hierarchical task planning for the apple harvesting robot is developed and implemented successfully. Future works in real-time implementation will focus on the replanning approach by using parallel programming for motion planner. In this approach, the motion of the arm from one position to target will be planned and re-planned several time while new data for the collision mapping are acquired until reaching the target. It requires to save the joint trajectory output of motion planning and only send to execute level a part of the output until a new update appears. The reliability and potential for speedups of the framework shows that it will be proper for the task executed in occluded and uncertain environment. Moreover, based on the architecture of task planning framework, autonomous planning for more complex actions without predefined program, e.g. guiding the arm to position that can have less occlusion while tracking the fruit, pushing the leaves away to grasp the fruit, etc. could be raised as issue for further investigation.

REFERENCES

J. Baeten et al. Autonomous Fruit Picking Machine: A Robotic Apple Harvester. In *6th International Conference on Field and Service Robotics - FSR 2007*, volume 42 of *Springer Tracts in Advanced Robotics*, Chamonix, France, 2007. Springer.

J. Baur, J. Pfaff, H. Ulbrich, and T. Villgrattner. Design and development of a redundant modular multipurpose agricultural manipulator. In *Advanced Intelligent Mechatronics (AIM), 2012 IEEE/ASME International Conference on*, pages 823 –830, 2012.

M. Conor et al. Model-based, hierarchical control of a mobile manipulation platform. In *ICAPS Workshop on Planning and Plan Execution for Real-World Systems*, Thessaloniki, Greece, 2009.

Y. Edan, D. Rogozin, T. Flash, and G.E. Miles. Robotic melon harvesting. *Robotics and Automation, IEEE Transactions on*, 16(6):831 –835, 2000.

Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013.

L.P. Kaelbling and T. Lozano-Perez. Hierarchical task and motion planning in the now. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1470 –1477, 2011.

Naoshi Kondo and K. C. Ting. *Robotics for Bioproduction Systems*. American Society of Agricultural Engineers, 1998.

T.Th Nguyen, D. Van Eessen, J. De Baerdemaeker, and W. Saeys. Optimum detaching movement for apples-harvesting robot. In *International Conference of Agricultural Engineering*, 2012.

S. Srinivasa, D. Ferguson, C. Helfrich, D. Berenson, A. Collet, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and M. Weghe. HERB: a home exploring robotic butler. *Autonomous Robots*, 28(1):5–20, 2010.

I.A. Sucan, M. Moll, and L.E. Kavraki. The open motion planning library. *Robotics Automation Magazine, IEEE*, 19(4):72 –82, 2012.

Z Xue, S. Ruehl, A. Hermann, T. Kerscher, and R. Dillmann. Autonomous grasp and manipulation planning using a tof camera. *Robotics and Autonomous Systems*, 60(3):387 – 395, 2012.

D. Zhao, J. Lv, W. Ji, Y. Zhang, and Y. Chen. Design and control of an apple harvesting robot. *Biosystems Engineering*, 110(2):112 – 122, 2011.