Cable Installation by a Humanoid Integrating Dual-arm Manipulation and Walking

Yili Qin[1,2], Adrien Escande[2] and Eiichi Yoshida[2,1]

***Abstract*— In this paper, we present a four layers hierarchical framework to complete the cable installation task with a humanoid robot. By decoupling task planning layer and motion planning layer, flexibility of the framework is revealed. After symbolic relational representation and planning method is used in task planning layer, the planning result is then mapped to primitive manipulation action sequence in motion planning layer. By integrating position detection of operating points and humanoid walking, installation of a long cable on two industrial clamps with a certain distance was completed by humanoid robot HRP-2Kai.**

I. INTRODUCTION

Deformable Linear Objects (DLOs), such as cables, hoses, leaf springs etc. are common in industrial production. On the other hand, large-scale manufacturing such as aerospace, ship and construction, most tasks are done manually by human workers. In contrast to highly automated automobile factory. The final assembly lines often require humans to carry out installation or assembly of cables, and this kind of work can be very hard and tedious (Fig. 1). It has a great economic significance to research how to use robots to support humans in deformable objects installation or assembly. Because of the cluttered nature of the environment, it is not possible to use fixed-based and wheel robots, and humanoid robots are good candidates. Their weakness of biped instability can be overcome by adding holding points with hands or arms in the environment. Moreover, humanoids can offer other benefits like easier adaptation to human environments and cooperation with humans.

Fig. 1: Airplane assembly including cable installation [1].

Despite those advantages, there are some difficulties in using the humanoid robot to handle the cable are as follows: (1) For motion planning of the robot, we need a correct model to predict the cable deformation. Since a cable is an object with large number of degrees of freedom, it is generally hard to predict its behavior very precisely. (2) We need to control the robot in such a way that the planned trajectory to the goal position should be followed to avoid unexpected interference with the environment. (3) Humanoid robots are not fixed at the floor unlike industrial robots. It is very important to keep balance of the humanoid robot to prevent falling when it is handling the cable.

DLO planning and manipulation problem is challenging due to the actuators have to handle a much larger number of uncontrollable degrees of freedom of DLO. Kavraki et al. discussed the path planning problem for an elastic object manipulated by two actuators, demonstrated the simulation result of positioning a metallic belt in a pipe assembly of a car by using their planner [17]. Bretl et al. showed simulations and experiments about quasi-static manipulation of a planar elastic rod by two and multiple robotic grippers [2][3]. Shah et al. formulated the manipulation planning problem for multiple inter-linked DLOs in aerospace manufacturing and presented an algorithm to avoid violation of interlink constraints [4]. However, those studies mainly focused on problem formulation or simulation under a fully controllable environment. For practical applications, their results need extension in terms of actuator dynamic control and robot displacement.

Complexity of the motion planning problem has an exponential relationship with dimension of the configuration space of robot. Because humanoid robot is with high degrees of freedom, it is hard to explicitly represent the configuration space. Sampling-based methods such as Rapidly exploring Random Tree (RRT) [7] and Probabilistic Roadmaps (PRM) [8] are widely used to solve this kind of problems. The early studies for humanoid robot whole-body motion planning is based on decoupled approach. The manipulation task is assigned to the upper body while the locomotion task is assigned to the lower body, we can find the related study in [9]. However, motions generated by using this method is not natural while humans use the coordinated action of whole body to complete a task. In [10], Yoshida et al. proposed a task-driven motion generation method. In this method, they employed generalized Inverse Kinematics (IK) to generate the whole-body motion including support polygon reshaping.

Extensive industrial exploration for DLO manipulation can be found starting around 2000. From 1999 to 2006, team of Dominik Henrich conducted intensive studies about manipulating a DLO. Their work includes: Contact states and states transition between a DLO and rigid environment [11]; Vibration reduction when quickly moving a DLO and

[1] University of Tsukuba, Japan.
[2] CNRS-AIST JRL (Joint Robotics Laboratory), UMI3218/RL, Tsukuba, Japan.

assembling it into a hole [13], and so on. In 2007, Saha et al. presented a new motion planner based on the topology model of DLO for typing knots with two cooperating industrial robotic arms [14]. In 2011, Jiang and Koo et al. conducted an engineering attempt to assemble a cable harness to the instrument panel frame of a car under a condition similar to an actual plant. But the total success rate is just 50% throughout the experiments [15].

However, efficient and flexible cable handling by humanoid robot has hardly been studied. We therefore investigate a whole-body motion planning and control method for handling DLOs by humanoid motions. In this paper, we present our research results that humanoid robot HRP-2Kai automatically installed a long cable on two industrial clamps (Fig. 2). The contributions of our work are as follows: By taking advantage of the recent progress on advanced whole-body control [5] and pattern generator [18], we present a general task description framework that can include robot walking and manipulation and moreover extended to other tasks.

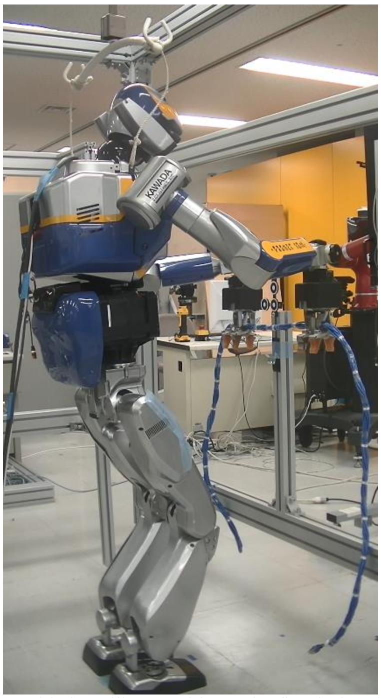


Fig. 2: HRP-2Kai is installing a cable on the first industrial cable clamp.

II. PROBLEM DESCRIPTION

Inspired the cable installation task in Fig. 1, we set the experiment scenario as Fig. 3 shows: We have a U-shape steel frame as the support structure. On each side of the support structure, there is a clamp used for fixing cable. Above each clamp, there is a marker used for locating the clamp's position. Due to there is a certain distance between two clamp positions. Humanoid robot need to move in front of the two clamps and complete cable installation.

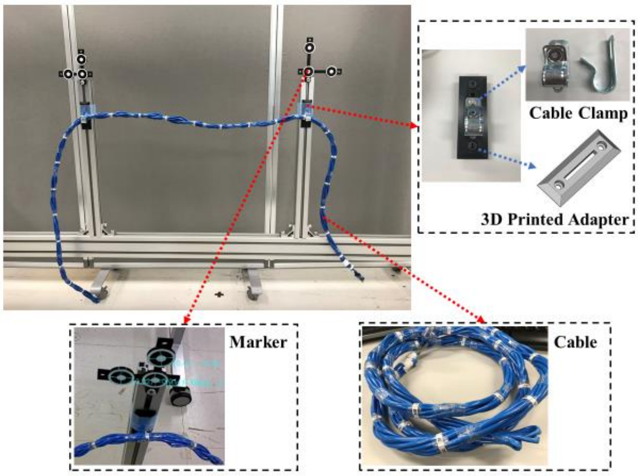


Fig. 3: The Experiment Set.

A. Task Components and Definitions

1) **Cable**

a) **Cable Shape:** The shape of the cable (Fig. 4) in 3D space can be described by a curve parameterized by the curvilinear length L. Each point on the cable can be denoted by a position $\mathbf{X}(s)$ and a local orientation $\mathbf{\Phi}(s)$ attached on it.

$$\mathbf{C}: s \in [0, L] \rightarrow \mathbf{C}(s) \in R^3 \times S^3 \qquad (1)$$

Additionally,

$$\begin{aligned} \mathbf{C}(s) &= [\mathbf{X}(s)\ \mathbf{\Phi}(\mathbf{s})] \\ \mathbf{X}(s) &= [x(s)\ y(s)\ z(s)]^T \\ \mathbf{\Phi}(s) &= [\emptyset(s)\ \theta(s)\ \varphi(s)]^T \end{aligned}$$

Where $\mathbf{C}(0)$ is referred to as the head of the cable, $\mathbf{C}(L)$ is referred to as its tail.

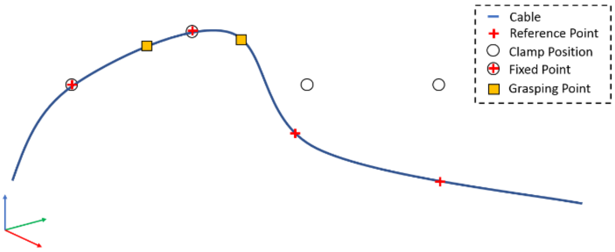


Fig. 4: Four layers of structure for cable installation task.

b) **Grasping Points:** A point on the cable, grasping by a gripper is referred to as a grasping point. When a point becomes a grasping point, this point will create a boundary condition for the cable shape $\mathbf{C}(s)$. The more grasping points on a cable are placed, the better controllability for the cable shape the robot will have. A set of grasping points can be defined as an order array:

$$\mathbf{G} = [g_j], g_j \in [0, L], j \in \{0,1, \dots, n_{\text{grasp}}\} \qquad (2)$$

c) **Reference Points:** A point on the cable will be installed on the according clamp is referred to as a reference point. A set of reference points can be defined as an order array:

$$\mathbf{R} = [r_j], r_j \in [0, \mathrm{L}], j \in \{0,1, \dots, \mathrm{n_{ref}}\} \tag{3}$$

d) **Fixed Points:** When a point on the cable is clamping by robot and fixed on the clamp, the point becomes a fixed point. Ideally, we expect all the reference points will become fixed points by robot manipulation. A set of fixed points can be defined as an order array:

$$\mathbf{F} = [f_j], f_j \in [0, \mathrm{L}], j \in \{0,1, \dots, \mathrm{n_{fix}}\} \tag{4}$$

2) **Clamp Positions**

The clamps are used to fix the cable. The reference points on the cable need to be clamped on the corresponding clamps. The clamp positions can be defined as:

$$\mathbf{K} = [K_j], K_j \in \mathrm{R}^3 \times \mathrm{S}^3, j \in \{0,1, \dots, \mathrm{n_{clamp}}\} \tag{5}$$

3) **Robot**

Configuration of the robot can be denoted by $\boldsymbol{q} \in \mathbf{Q}$. Where $\mathbf{Q}$ is configuration space.

4) **Predefined Information**

Due to the limitations of no sensing for the whole environment and for the shape of cable in 3D space. We need to specify some information for the environment (e.g. while handling the cable in front of each clamp, the feasible positions for the robot) and constraints for the robot (e.g. for each robot moving, the maximum permissible distance for the robot). Here, we denote all this information by **I**.

B. System States and Transition

We defined the state of the system by consideration the shape of the cable **C**, the grasping points **G** and the fixed points **F** on the cable, the clamp positions **K**, configuration of the robot $\boldsymbol{q}$. The state of the system can be described as the follow tuple:

$$\mathbf{S} = (\mathbf{C}, \mathbf{G}, \mathbf{F}, \mathbf{K}, \boldsymbol{q}) \tag{6}$$

Assume the action acting on current system state is $\mathrm{A_n}$, the system state transition can be represented as:

$$\mathbf{S}_{\mathrm{n+1}} = \mathbf{S}_{\mathrm{n}}(\mathrm{A_n}) \tag{7}$$

C. System State Transition Actions

When manipulating an object, according to the different shapes, structures and other characteristics of the object, people usually choose different and relatively fixed set of actions. For example, when we are handling a rope and want to extend the length of the rope between two hands, we usually choose "spread" action. To achieve this action, we use one hand to fix the rope, then open two arms and let the rope slide in the other hand. Inspired cable manipulation by human motions, we predefined 10 kinds of executable primitive manipulation actions for humanoid robot. Each primitive action can receive some parameters to adapt to changes in tasks and installation environment. their definitions are as follows:

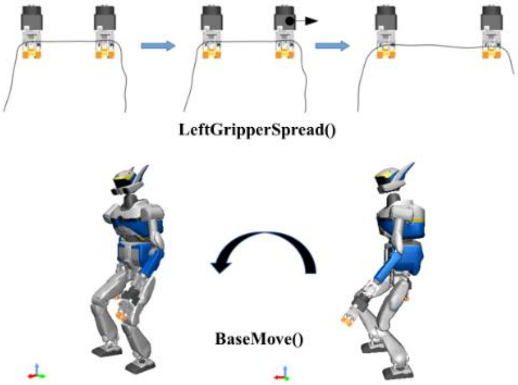


Fig. 5: Examples for primitive actions.

(1) **LeftGripperSpread(**leftTarget**)** - Robot uses right gripper to fix the cable, then uses left gripper to spread the cable. By using "spread" action type, robot can change the grasping point on a cable (Fig. 5). Parameter leftTarget $\in \mathrm{R}^3 \times \mathrm{S}^3$.

(2) **LeftGripperRegrasp(**leftTarget**)** - Robot re-grasps the cable with left gripper. By using "re-grasp" action type, robot can add a grasping point for the cable. Parameter leftTarget $\in \mathrm{R}^3 \times \mathrm{S}^3$.

(3) **LeftGripperRelease()** - Robot releases the grasping point in left gripper.

(4) **RightGripperSpread(**rightTarget**)** - Robot uses left gripper to fix the cable, then uses right gripper to spread the cable. Parameter rightTarget $\in \mathrm{R}^3 \times \mathrm{S}^3$.

(5) **RightGripperRegrasp(**rightTarget**)** - Robot re-grasps the cable with right gripper. Parameter rightTarget $\in \mathrm{R}^3 \times \mathrm{S}^3$.

(6) **RightGripperRelease()** - Robot releases the grasping point in right gripper.

(7) **BaseMove(**baseTarget**)** - Robot moves to goal position and orientation by walking with cable in one of the grippers (Fig. 5). Parameter baseTarget $\in \mathrm{R}^2 \times \mathrm{S}^1$.

(8) **InstallCable()** - Robot uses vision to detect the marker position and calculate the clamp position, then installs the cable on the clamp with two grippers. By using "install" action type, robot can add a fixed point for the cable.

(9) **TwoGripperGrasp(**leftTarget, rightTarget**)** - Robot grasps a cable on the support structure with two grippers. Parameter leftTarget $\in \mathrm{R}^3 \times \mathrm{S}^3$ and rightTarget $\in \mathrm{R}^3 \times \mathrm{S}^3$.

(10) **TwoGripperRelease()** - Robot opens two grippers and releases the cable.

D. Plan Structure

As an intermediate result, the plan of the task planning layer is consisted of symbolic actions and can be represented by the sequence as follows:

$$\mathbf{P_t} = (A_{t1}, A_{t2}, \dots, A_{tn}) \tag{8}$$

The plan of the motion planning layer is the final result that consisted of primitive manipulation actions and can be represented by the sequence as follows:

$$\mathbf{P_m} = (A_{m1}, A_{m2}, \dots, A_{mn}) \tag{9}$$

III. CABLE INSTALLATION TASK PLANNING

As shown in Fig.6, separating strategy is using between task planning layer and motion planning layer. This decoupled approach brings the merit that we can just easily modify the motion planning layer and use other kinds of robot (e.g. a dual-arm robot or field mobile robot) to complete the same cable installation task.

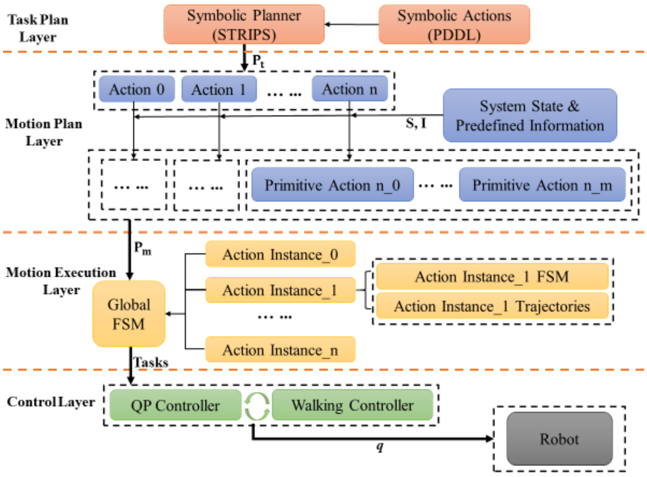


Fig. 6: Four layers of structure for cable installation task.

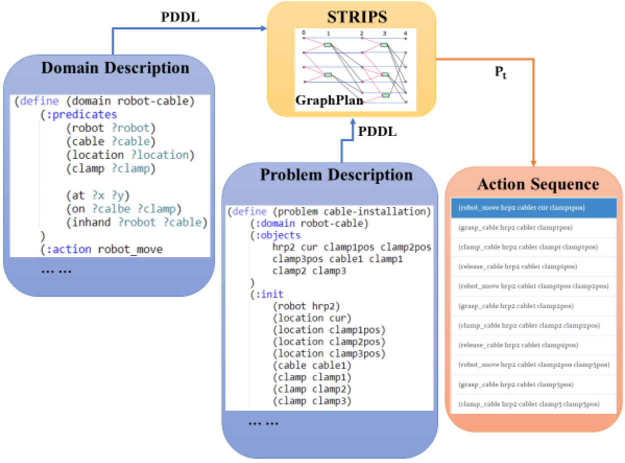
Task planning identifies sequence of actions that transit the system from initial state to a goal state. In task planning layer, we use symbolic relational representation and planning method for cable installation problem. Stanford Research Institute Problem Solver (STRIPS) is an automated planner using GraphPlan algorithm for automated planning and scheduling in artificial intelligence. The pioneering work on STRIPS can be found in [16]. As Fig. 7 shows, by using Planning Domain Definition Language (PDDL) and describing the planning domain and planning problem respectively for a cable installation task, we can use STRIPS to plan and generate a sequence of actions. Four kinds of symbolic actions for cable installation task are predefined by us:

(1) **robot_move(**r, x, y**)** - Robot r moves from location x to location y.

(2) **grasp_cable(**r, ca, x**)** - Robot r grasps cable ca at location x.

(3) **clamp_cable(**r, ca, cp, x**)** - At location x, robot r clamps cable ca on clamp cp.

(4) **release_cable(**r, ca, x**)** – At location x, Robot r releases cable ca.



```
(define (domain robot-cable)
    (:predicates
        (robot ?robot)
        (cable ?cable)
        (location ?location)
        (clamp ?clamp)

        (at ?x ?y)
        (on ?calbe ?clamp)
        (inhand ?robot ?cable)
    )
    (:action robot_move
```

```
(define (problem cable-installation)
    (:domain robot-cable)
    (:objects
        hrp2 cur clamp1pos clamp2pos
        clamp3pos cable1 clamp1
        clamp2 clamp3
    )
    (:init
        (robot hrp2)
        (location cur)
        (location clamp1pos)
        (location clamp2pos)
        (location clamp3pos)
        (cable cable1)
        (clamp clamp1)
        (clamp clamp2)
        (clamp clamp3)
```

```
(robot_move hrp2 cable1 cur clamp1pos)
(grasp_cable hrp2 cable1 clamp1pos)
(clamp_cable hrp2 cable1 clamp1 clamp1pos)
(release_cable hrp2 cable1 clamp1pos)
(robot_move hrp2 cable1 clamp1pos clamp2pos)
(grasp_cable hrp2 cable1 clamp2pos)
(clamp_cable hrp2 cable1 clamp2 clamp2pos)
(release_cable hrp2 cable1 clamp2pos)
(robot_move hrp2 cable1 clamp2pos clamp3pos)
(grasp_cable hrp2 cable1 clamp3pos)
(clamp_cable hrp2 cable1 clamp3 clamp3pos)
```

Fig. 7: Solve cable installation problem in task planning layer by using PDDL and STRIPS.

For a cable installation task, in domain description we first defined the objects (e.g. cable, robot, clamp, and so on) and their relations (e.g. on(cable, clamp) represents a cable is on a clamp). Then we defined the actions which can change the relations between objects. In problem description, we described the initial state and goal state of the robot and the cable. STRIP then creates a graph by using domain description and gets a plan according problem description.

IV. CABLE INSTALLATION MOTION PLANNING

A. From Symbolic Actions to Primitive Actions

To complete a specific cable installation task, a connected path in configuration space of the robot is necessary. Unlike the function of task planning layer which is focus on the abstraction for the task, the motion planning layer should actually generate some actions that can be executed by robot. In section II.*C*, 10 kinds of primitive actions have been defined. As shown in Fig. 6, the motion planning layer takes sequence of symbolic actions as input. By considering the system state **S** and predefined information **I**, this layer will map each symbolic action to one or several primitive actions and generate the parameters for each primitive action. The output will be a sequence of primitive actions with their parameters.

As a simple example, we assume that the predefined maximal movable distance for the robot is 0.60m each time and the plan from task planning layer just includes one symbolic action:

(robot_move(hrp2, clamp1_pos, clamp2_pos))

This means current position of robot hrp2 is in front of clamp1. Robot hrp2 need move to the position in front of clamp2. In information **I**, we can get the distance between two clamps and their relative direction. If clamp2 is 0.50m left of clamp1, this symbolic action will be decomposed into three primitive actions:

(LeftGripperSpread(), RightGripperRelease(), BaseMove())

The robot just needs to move one time. Their parameters can also be generated by considering system state $\boldsymbol{S}$ and information $\mathbf{I}$. If clamp2 is 0.8m left of clamp1, distance 0.8m will be divided into 0.6m and 0.2m. The robot needs to move two times to arrive at the target position.

B. Implementation of Primitive Actions

Motion execution layer has a two layers of Finite State Machine (FSM) structure. A FSM is a model used to represent and control execution flow. It can produce great results without a complex code [19]. The global FSM invokes the action instances according to the action sequence. Each primitive action has its own local FSM to drive the control logic. Since a humanoid robot has high degrees of freedom, generating stable and safe motions is already a challenge. To generate motions for primitive manipulation actions, we have manually designed a series of stable key postures without changing the foot positions. By checking whether the projection point of robot's Center of Mass (CoM) is inside the area of support polygon defined as a convex hull of the feet, we can judge stability of the posture. Here we take the key postures as the states of local FSM, use the gripper's position and state to control the FSM to jump from one state to another. With this method, we can get a set of movements of robot for cable handling as shown in Fig. 8. We can also control the robot to stop running at any state by simply sending an ending signal to the FSM. The output motion sequence is then sent to the control layer described in the next section.

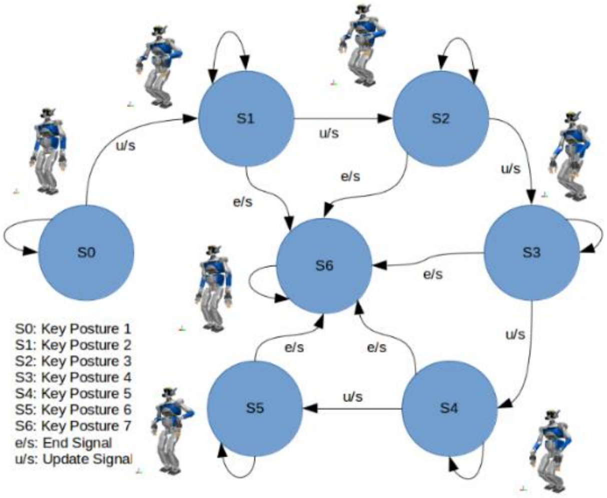


Fig. 8: Implementation for TwoGripperGrasp() primitive action by using key postures and FSM.

V. QP-BASED CONTROLLER

In control layer, we use two controllers. A multi-objective Quadratic Programing (QP) controller [5] is used to generate whole-body motions for cable installation. The other controller is used for humanoid walking, details can be found in [10].

Since humanoid robot is a kind floating base system, the controller needs to complete multiple tasks at the same time (e.g. CoM task for balance, position task for a body, and so on). The tasks are formulated as a QP problem and solved by using the LSSOL QP solver [6]. As a simple case, Fig. 9 shows the diagram of the controller for a single end effector task. This task need robot to move one hand from current position to a goal position. The controller computes the acceleration $\ddot{\boldsymbol{q}}$ for each joint at each time step Δt . Formulation for this task is as follows:

$$\begin{aligned} &\underset{\ddot{q},\tau,f}{\boldsymbol{min}.} \left\| \boldsymbol{J_h}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J_h}}\dot{\boldsymbol{q}} - \ddot{\boldsymbol{e}}_{\boldsymbol{h}}^{*} \right\|^{\boldsymbol{2}} \\ \textit{s.t.}\quad &\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{N}(\boldsymbol{q},\dot{\boldsymbol{q}}) + \boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{L_\tau} + \boldsymbol{J_c^T f} \\ &\boldsymbol{\tau}^- \leq \boldsymbol{\tau} \leq \boldsymbol{\tau}^+, \boldsymbol{Cf} \geq \boldsymbol{0} \\ &\ddot{\boldsymbol{q}}^- \leq \ddot{\boldsymbol{q}} \leq \ddot{\boldsymbol{q}}^+ \\ &\boldsymbol{J_f}\ddot{\boldsymbol{q}} + \dot{\boldsymbol{J}}_{\boldsymbol{f}}\dot{\boldsymbol{q}} = \boldsymbol{0} \end{aligned} \tag{10}$$

$\boldsymbol{q}$ **-** Configuration of the robot.
$\boldsymbol{\tau}$ **-** Joint torques.
$\boldsymbol{e_h}$ **-** Effector position task defined by error. $\boldsymbol{e_h} = x_h(\boldsymbol{q}) - x_h^d$.
x_h **-** Position of the end effector.
x_h^d **-** Goal position for the end effector task.
$\boldsymbol{e_h^*}$ **-** Task goal specified by us.
$\boldsymbol{J_h}$ **-** Jacobian matrix of the end effector.
$\boldsymbol{J_c}$ **-** Jacobian matrix of the contact points.
$\boldsymbol{J_f}$ **-** Jacobian matrix of the feet.
$\boldsymbol{L_\tau}$ **-** Selection matrix accounting for the under-actuation of the robot base.
$\boldsymbol{M}$ **-** Inertia matrix of the robot.
$\boldsymbol{N}$ **-** Coriolis.
$\boldsymbol{g}$ **-** Gravity.
$\boldsymbol{C}$ **-** Discretized Coulomb friction cones matrix.
$\boldsymbol{f}$ **-** A set of contact forces.

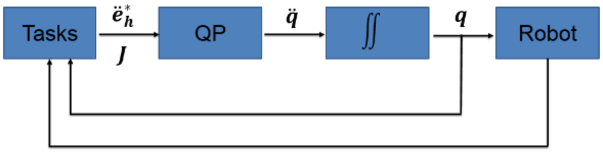


Fig. 9: Quadratic Programing (QP) based controller for whole-body motion generation

VI. EXPERIMENTAL RESULTS

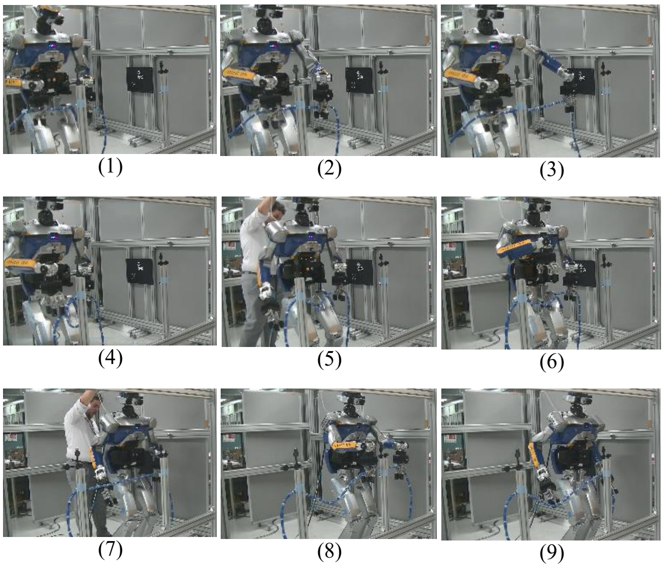
We use the Whycon library [12] for marker detection. Length of the cable is 2.40m. Each 10cm, we attached a label to indicate the position. Weight of the cable is 0.35kg. Distance between two clamps is 0.80m. Hight of the two clamps' positions is 0.85m.

We specified the information such as robot initial position, number of clamps and their positions etc. for task planner and motion planner. By using off-line computation, we got a plan which was then used to do the cable installation experiment. Figure 10 shows the snapshot of the experiment. The robot was putted on the initial position in front of the first clamp (about 0.40m) with a cable in its two grippers. By using vision to detect the marker position, robot installed the cable on the clamp. Then robot spread the cable to get more space and moved to the second clamp. After two times of spreading and moving, the robot arrived the position in front of the second clamp and continued to install the cable on the clamp. The experimental results show that the proposed approaches are practical and effective.

VII. SUMMARY AND FUTURE WORK

In this paper, we presented our work about cable installation between two industrial clamps with a humanoid robot. Since we expect our framework has capacity of general task description and can be easily used in other tasks, we proposed a decoupled approach for task planning layer and motion planning layer. By using an integration method including marker detection by vision, walking controller and QP-based whole-body motion controller, we completed the experiment. To the best of our knowledge, there are no other work before trying to combine position detection and robot displacement while handling a cable. We also found some problems during the experiments. When the cable was manipulated by the high rigidity robot system, behavior of the cable become harder to predict due to the cable torsion. Sometimes “re-grasp” action for the cable failed and robot lost control for the cable. A precise cable model and some parameter identification methods may be helpful to tackle this problem. For practical industrial applications, we should add more error handling mechanisms to prevent motion failures.

Although the feasibility has been confirmed by the experiments, the method still has room to improve for more autonomous and versatile cable manipulation. The future issues include: (1) Apply perception of environment and cable shape in 3D space including vision and other sensors, and the motion planning to allow the robot to handle the cable automatically. (2) Develop a general motion planning method by integrating an accurate cable model predicting its behavior.


(1) (2) (3)

(4) (5) (6)

(7) (8) (9)

Fig. 10: Experiment results.

REFERENCES

[1] Bolotnikova A, Chappellet K, Paolillo A, et al. A circuit-breaker use-case operated by a humanoid in aircraft manufacturing[C]//Automation Science and Engineering (CASE), 2017 13th IEEE Conference on. IEEE, 2017: 15-22.

[2] Matthews, Dennis, and Timothy Bretl. "Experiments in quasi-static manipulation of a planar elastic rod." Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.

[3] Mukadam, Mustafa, Andy Borum, and Timothy Bretl. "Quasi-static manipulation of a planar elastic rod using multiple robotic grippers." Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on. IEEE, 2014

[4] Shah, Ankit J., and Julie A. Shah. "Towards manipulation planning for multiple interlinked deformable linear objects." Robotics and Automation (ICRA), 2016 IEEE International Conference on. IEEE, 2016.

[5] Pfeiffer K, Escande A, Kheddar A. Nut fastening with a humanoid robot[C]//Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on. IEEE, 2017: 6142-6148.

[6] Gill P E, Hammarling S J, Murray W, et al. User's guide for LSSOL (Version 1.0): A Fortran package for constrained linear least squares and convex quadratic programming[M]. Systems Optimization Laboratory, Department of Operations Research, Stanford University, 1986.

[7] LaValle, S., Kuffner, J.: Rapidly-exploring random trees: Progress and prospects. In: K.M. Lynch, D. Rus (eds.) Algorithmic and Computational Robotics: New Directions, pp. 293–308. A K Peters (2001)

[8] Kavraki, L., Svestka, P., Latombe, J.C., Overmars, M.: Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Trans. on Robotics and Automation 12(4), 566–580 (1996)

[9] Yoshida, E., Esteves, C., Belousov, I., Laumond, J.P., Sakaguchi, T., Yokoi, K.: Planning 3D collision-free dynamic robotic motion through iterative reshaping. IEEE Trans. on Robotics 24(5), 1186–1198 (2008). DOI 10.1109/TRO.2008.2002312

[10] Yoshida E, Kanoun O, Esteves C, et al. Task-driven Support Polygon Reshaping for Humanoids[C]//Humanoids. 2006: 208-213.

[11] Remde, Axel, Dominik Henrich, and H. Wom. "Manipulating deformable linear objects-Contact state transitions and transition conditions." Intelligent Robots and Systems, 1999. IROS'99. Proceedings. 1999 IEEE/RSJ International Conference on. Vol. 3. IEEE, 1999.

[12] Nitsche M, Krajnik T, Cizek P, et al. WhyCon: an efficient, marker-based localization system[J]. 2015.

[13] Yue, Shigang, and Dominik Henrich. "Manipulating deformable linear objects: model-based adjustment-motion for vibration reduction." (2001).

[14] Saha, Mitul, and Pekka Isto. "Manipulation planning for deformable linear objects." IEEE Transactions on Robotics 23.6 (2007): 1141-1150.

[15] Koo, Kyongmo, et al. "Development of a wire harness assembly motion planner for redundant multiple manipulators." Journal of Robotics and Mechatronics 23.6 (2011): 907.

[16] R. E. Fikes and N. J. Nilsson, “STRIPS: A new approach to the application of theorem proving to problem solving,” Artificial intelligence, vol. 2, no. 3, pp. 189–208, 1972.

[17] Lamiraux, Florent, and Lydia E. Kavraki. "Planning paths for elastic objects under manipulation constraints." The International Journal of Robotics Research 20.3 (2001): 188-208

[18] Harada K, Miura K, Morisawa M, et al. Toward human-like walking pattern generator[C]//Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on. IEEE, 2009: 1071-1077.

[19] Vaillant J, Kheddar A, Audren H, et al. Multi-contact vertical ladder climbing with an HRP-2 humanoid[J]. Autonomous Robots, 2016, 40(3): 561-580.