# Elastic Strips: Implementation on a Physical Humanoid Robot

Jinsung Kwon, Taizo Yoshikawa, and Oussama Khatib

***Abstract*— For robots to operate in human environments, they are required to react safely to unexpected changes in the work area. However, existing manipulation task planning methods take more than several seconds or minutes to update their solutions when environmental changes are recognized. Furthermore, the computation time exponentially increases in case of highly complex structures such as humanoid robots. Therefore, we propose a reactive system for high d.o.f. robots to perform interactive manipulation tasks under real-time conditions. The paper describes the implementation of the Elastic Strip Framework, a plan modification approach to update initial motion plans. To improve its real-time performance and reliability, the previous geometric approximation is replaced by an implicit method that constructs an elastic tunnel for collision checking. Additionally, in order to maintain a robust system even in exceptional situations, such as undetected obstacles, the force transformer module executes compliant motions, and the current elastic strip adapts the path tracking motion by monitoring tracking errors of the actual motion. The proposed system is applied to a Honda humanoid robot. Real-time performance is successfully demonstrated in real-world experiments.**

## I. INTRODUCTION

Robot work places have expanded from isolated environments such as factories to human environments, where every surrounding object can change its position without a deterministic schedule, which is called a dynamic environment. Therefore, in order to satisfy demands for future roles of robots in human environments, it becomes necessary to develop robot planning and control systems that handle complex manipulation tasks in dynamic environments.

This paper presents a planning and control system, which can satisfy such requirements. We report about real-world experimental results of a humanoid robot executing multiple reactive manipulation tasks simultaneously. The system presented in this paper has following features: i) dynamically changing task objectives; ii) avoiding collisions with unpredictably moving obstacles; iii) handling multiple manipulation tasks within one constraint-consistent control framework; and iv) robust motions even during exceptional events, such as contacts with undetected objects.

Motion planning for manipulation tasks was studied for static environments by off-line configuration-space based planners [1], [2]. An on-line manipulation task planner was developed for motions with two robot arms [3]. Probabilistic roadmap-based approaches were also proposed for manipulation planning systems [4], [5]. Several methods for finding collision-free manipulation motions for animated human characters [6], [7] or humanoid robots [8] were proposed. However, the previous motion planning methods require computation times ranging from several seconds up to the order of minutes in order to find a solution of a given manipulation task problem — which is not fast enough for real-time applications. In addition, in case of complex kinematic systems, it is computationally expensive to update in real-time configuration space obstacles in high dimensional spaces. Therefore, in dynamically changing environments, the use of global motion planning methods is limited to simpler conditions such as slowly changing environments or simple manipulation tasks.

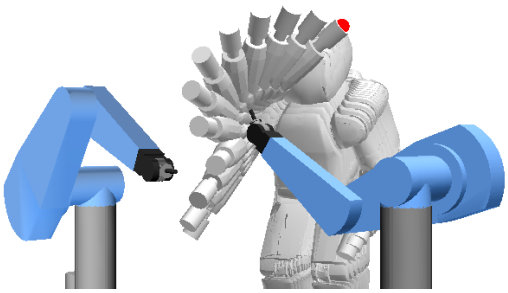

Fig. 1: The Elastic Strips framework maintains a collision-free motion plan in dynamically changing environments. In the figure, a humanoid robot can reach a goal position (red) without colliding the dynamically moving obstacles (the blue robots).

Since whole-body motion planning and real-time control is a challenging task, diverse approaches have been proposed and tested in different aspects. In [9], [10], a whole-body impedance-based controller enables the humanoid Justin to perform tasks compliantly while providing a low-dimensional task space and reactive self-collision avoidance motion. In [11], a two-armed and two-wheeled mobile robot is controlled for whole-body manipulator tasks by generalized inverse dynamics. A safe human-robot cooperation is also enabled by an integrated control framework [12], [13], which realizes a reactive collision avoidance, a safe physical contact interaction, and a joint limit avoidance. In [14], human-robot interaction is realized by enabling a humanoid to compliantly follow physical contact forces of a human. In [15], a hybrid motion planning method is proposed which combines a sampling-based planning algorithm in the low dimensional task space and a reactive local avoidance motion in the null space. Real-time self-collision avoidance method [16] is also proposed for whole-body motion of humanoids. In [17], a manipulation planning framework based on task

Jinsung Kwon and Oussama Khatib are with Artificial Intelligence Laboratory, Stanford University jinskwon@stanford.edu, ok@cs.stanford.edu

Taizo Yoshikawa is with Honda Research Institute USA, inc. tyoshikawa@hra.com, taizouy@stanford.edu

space regions is proposed for manipulation tasks which have constraints on end-effector poses.

In order to maintain a valid solution path in changing situations, the Elastic Band method [18] modifies configuration-space paths by using proximity information of the work-space in real-time. This concept of path modification was expanded to more complex robots by the Elastic Strips method [19], which updates task consistent motion plans directly in the robot's workspace. Moreover, Elastic Roadmaps [20] combine the control-based modification method and the global connectivity roadmap method. In this paper, we aim to apply a modified version of the Elastic Strips approach to a Honda humanoid robot platform in order to demonstrate reactive behaviors during manipulation tasks in real-time while the robot's environment changes dynamically without prior knowledge.

## II. SYSTEM ARCHITECTURE

The system consists of the Elastic Strip module, a motion interpolation module, and a robot controller. Additional elements include sensors and a communication network between the modules. In order to maintain a valid solution for a task motion in real time, the Elastic Strip module modifies an initial global motion plan to adapt to environment changes detected by sensors. The motion interpolation module extracts continuous task motion commands from the elastic strip path and sends the commands to the robot controller. The robot controller produces robot motion by producing motor torques at the lowest level. The low level control framework is constructed with the Honda locomotion controller for the lower body leg motion and the Stanford whole body control framework for the upper body manipulation task motion [21]. The lower body controller maintains a stable leg posture for the body balance and produces a desired hip position and orientation which are commanded by the planning layer. The whole body control framework enables upper body manipulators to execute multiple task objectives. The force transformer algorithm can also be integrated in the system in order to enable a position control-based robot to run by compliant force control. The proposed system is described in Figure (2).

### A. Elastic Strip Framework

The Elastic Strip framework is an approach for the integration of global behavior and local avoidance behavior to enable reactive motion execution while maintaining the global properties of the task. This is accomplished by an incremental modification of the previously planned motion. The modification approach preserves topological properties of the original path as well as task constraints. Thus, it can maintain a task-consistent valid path from the current state to the desired goal state without being susceptible to local minima.

An initial candidate path is provided by a global motion planner and the Elastic Strip framework constructs nodes with robot configurations along the path. At the beginning, two nodes are constructed at the start and end positions of the path. Then, it keeps creating new nodes recursively until the collision free motion is guaranteed by the collision checking algorithm along the initial path.

Let $\mathscr{P}_c$ be a *candidate path* representing a collision-free path motion accomplishing a given task. Then, $V_{\mathscr{P}_c}$ is defined as the work space volume swept by the robot. An elastic strip $\mathscr{S}$ is defined as $\mathscr{S} = (\mathscr{P}_c, \mathscr{T}_{\mathscr{P}_c})$. $\mathscr{T}_{\mathscr{P}_c}$ is the *elastic tunnel*, which is a work space volume of free space surrounding $V_{\mathscr{P}_c}$. Finding an efficient method for computing in real-time $\mathscr{T}_{\mathscr{P}_c}$ is very critical in building a reactive framework. The condition for the candidate path to be collision-free is

$$V_{\mathscr{P}_c} \subseteq W \setminus V_{\mathscr{O}} \tag{1}$$

where $W$ is the overall work space and $V_{\mathscr{O}}$ is the work space occupied by obstacles. To prove that the collision-free condition in (1) holds for a candidate path, it is required to provide a valid elastic tunnel $\mathscr{T}_{\mathscr{P}_c}$ satisfying

$$V_{\mathscr{P}_c} \subseteq \mathscr{T}_{\mathscr{P}_c} \tag{2}$$

and

$$\mathscr{T}_{\mathscr{P}_c} \subseteq W \setminus V_{\mathscr{O}} \tag{3}$$

The candidate path $\mathscr{P}_c$ is represented as a discrete set of consecutive configurations, called a *set of nodes*. Virtual robots at these nodes are exposed to forces in the work space, to incrementally modify the candidate path to yield a new one. Real-time performance can be achieved by using a potential field-based control algorithm for the modification forces without replanning the whole path. Rather than exploring the entire configuration space, the algorithm gathers proximity information to the environment and finds a new candidate path directly in the work space by using the kinematic structure of the robot. In the framework, each node provides local information at the configuration and continuous feasibility to the goal is provided by a segment between adjacent nodes which keeps monitoring free space by maintaining a valid elastic tunnel $\mathscr{T}_{\mathscr{P}_c}$. Here, the nodes provide boundary conditions to the segments to construct continuous elastic tunnels.

The modification forces are derived from two potential functions, the external and internal potential, $V_{ext}$ and $V_{int}$, respectively. The external, repulsive potential $V_{ext}$ is for maximizing the clearance the path has against obstacles. For a point $p$ on a robot body of a node, the external potential field is defined as

$$V_{ext}(p) = \begin{cases} \frac{1}{2} k_r (d_0 - d(p))^2 & \text{if } d(p) < d_0 \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $d(p)$ is the distance from $p$ to the closest obstacle, $d_0$ defines the influence region around obstacles, and $k_r$ is the repulsion gain. The resulted repulsive force acting on $p$ to push the trajectory away from obstacles is

$$F_p^{ext} = -\nabla V_{ext} = k_r (d_0 - d(p)) \frac{d}{\|d\|} \tag{5}$$

where $d$ is the vector between p and the closest point on the hull of an obstacle.

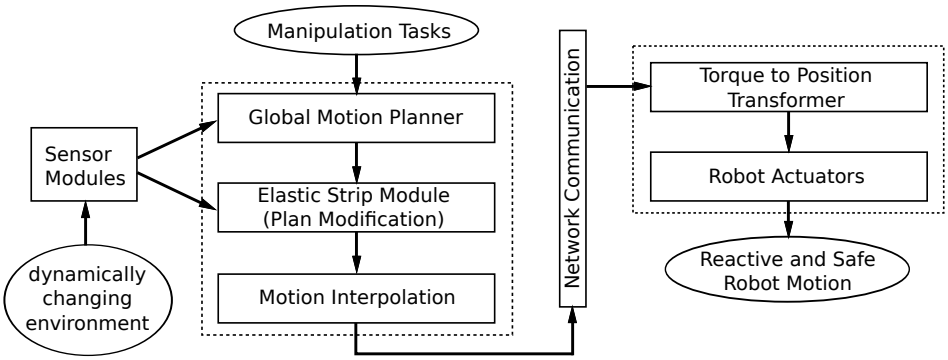


Fig. 2: System architecture: The planning layer is composed of a global motion planner, the Elastic Strip, and the motion interpolation module. The force transformer interprets a force command as a position/velocity command to realize compliant force control in a position-based control system.

To shorten the path when obstacles move away from the currently deformed path, virtual springs are attached at control points on consecutive robot models along the elastic strip. Let $p_j^i$ be the position vector of the control point attached to the $j$th link at the $i$th node. The internal contraction force acting at $p_j^i$ is defined as

$$F_{i,j}^{int} = k_c \left( \frac{d_j^{i-1}}{d_j^{i-1} + d_j^i} (p_j^{i+1} - p_j^{i-1}) - (p_j^i - p_j^{i-1}) \right) \quad (6)$$

where $d_j^i$ is the distance $\|p_j^i - p_j^{i+1}\|$ and $k_c$ is the contraction gain. To prevent the tension from pushing a node to obstacles, the internal forces in the elastic strip is computed by the local curvature of the strip rather than by the distance between nodes.

However, the external and internal forces may result in undesired or unnatural motion behaviors. This is avoided by imposing an additional posture potential, which describes a preferred posture, keeps joint angles in desired ranges, or maintains balance. For example, the total center of mass can be controlled to stay at a preferred position with a posture energy function, and corresponding torque

$$V_{posture}(q) = \frac{1}{2} k (x_{CoM}^2 + y_{CoM}^2) \quad (7)$$

$$\Gamma_{posture} = J_{CoM}^T (-\nabla V_{posture}) \quad (8)$$

where $k$ is a constant gain, $x_{CoM}$ and $y_{CoM}$ are the $x$ and $y$ offsets of the center of mass from the desired position, and $J_{CoM}$ is Jacobian matrix of the center of mass.

To ensure execution of the behavior with highest priority, the *whole body control framework* is applied to the robot behavior control at each node [21], [22]. In case of a redundant robot, the task execution and posture behavior can be performed simultaneously. The overall motion behaviors consist of three components, which are task, constraint, and posture. Constraints represent critical requirements of the motion such as obstacle avoidance, joint limits and self-collision avoidance. The priority consistent control torque is computed as

$$\Gamma = \Gamma_{constraint} + N_{constraint}^T (\Gamma_{task} + N_{task}^T \Gamma_{posture}) \quad (9)$$

where $\Gamma_{constraint}$, $\Gamma_{task}$, $\Gamma_{posture}$ are torques for constraint, task, and posture behaviors respectively, and $N_{constraint}$, $N_{task}$ are null space matrices of constraint and task respectively. In the control structure, constraint behavior has the highest priority to avoid fatal violation of the constraints. And the posture control is performed in the null space of the task in order not to affect the task execution. The null space matrices are defined as

$$N_{constraint} = I - J_{constraint}^T \bar{J}_{constraint}^T \quad (10)$$

$$N_{task} = I - J_{task}^T \bar{J}_{task}^T \quad (11)$$

where $\bar{J}$ is a dynamically consistent generalized inverse [23], which minimizes the kinetic energy,

$$\bar{J}^T = \Lambda J A^{-1} \quad (12)$$

where $A$ is the $n \times n$ kinetic energy matrix in the joint space and

$$\Lambda = (J A^{-1} J^T)^{-1} \quad (13)$$

is the kinetic energy matrix in the operational space.

The collision avoidance force $F_p^{ext}$ is performed in the posture torque $\Gamma_{posture}$ when there is enough margin to the closest obstacle, to produce *task-consistent* collision avoidance behavior. However, if an obstacle approaches very close to the configurations, $F_p^{ext}$ is performed as a constraint to avoid fatal collisions. Likewise, to ensure task-consistent behaviors, any constraint conditions can only be activated when violation is anticipated. The internal contraction force $F_{i,j}^{int}$ is performed also as posture behavior to produce tension in a task-consistent manner.

### *B. Motion Interpolation*

The elastic strip consists of discrete robot motions at nodes. However the real robot does not perform the motion of the virtual robots but moves *through* these configurations *along the elastic strip*. To move along the elastic strip, each segment has an interpolation equation whose boundary conditions are provided by states of discrete configurations at both ends. The resulting interpolated motion moves inside the

elastic tunnel to reach the goal state. To produce a tracking motion of the path, a virtual robot model is controlled by

$$\Gamma = \Gamma_{tracking\ task} + N^T_{tracking\ task}\Gamma_{posture} \tag{14}$$

The tracking control does neither include an external repulsive force nor an internal contraction force since the tracking control can satisfy the collision avoidance for itself.

### C. Compliant Force Control

In the hardware control, a compliant force control approach is adapted for safe and stable robot motion in rapidly changing human environments. In order to realize compliant force control, the system should be able to produce desired force at each actuator. However, most of the existing robots including the Honda humanoid robot are designed as a position-based control system. Thus, the force transformer method [24], [25] is implemented on the actuator control loop to produce desired torque. The force transformer uses desired torque and actual motor state to compute motor commands by the following equation:

$$\dot{q}_{cmd} = \frac{\tau_{cmd}}{K_v \cdot K_t \cdot CL} + \dot{q}_{act} \tag{15}$$

where $K_v$, $K_t$, and $CL$ are the velocity gain, the motor torque constant, and the transfer function of the current feedback loop, respectively. The transformer equation (15) computes the velocity control command, $\dot{q}_{cmd}$, from the desired torque, $\tau_{cmd}$, and the actual velocity, $\dot{q}_{act}$. The force transformer enables a position-based control system to control torque without hardware modifications.

## III. COMPLICATIONS OF MANIPULATION TASK BY HIGH D.O.F. ROBOTS

Due to the increased complexity of high d.o.f. robot such as a humanoid, it is very challenging to achieve real-time performance in manipulation tasks even with the elastic strip plan modification approach. For each link, the actual motion trajectory in the work space becomes more and more complicated as the number of joints involved increases. For example, in case of humanoids, it is extremely expensive to compute a tight boundary around the swept volume $V_{\mathscr{P}_c}$ of every link body due to the complexity and the large number of bodies. To improve performance, simplified geometric approaches can be considered in finding an elastic tunnel $\mathscr{T}_{\mathscr{P}_c}$. But this could also result in a false negative errors when collision checking is done. This would severely threaten the performance and safety of the system. Therefore, for real-time motion modification of complex robots , the computational load and reliability of a collision checking algorithm should not be much affected by the numbers of joints or linkages or by the complexity of each body motion.

The system proposed in this paper is more focused on providing a real-time algorithm for a manipulation task in narrow spaces rather than a path planning task in large spaces. Thus, in our target task behaviors, the base motion is assumed to be limited to small ranges, and the configurations of manipulators are assumed to have large changes during the task motion. In such applications, it is important to maintain connectivity between diverse configurations in a changing environment. Especially, the collision checking algorithm should be able to detect collisions in complex manipulation motions without compromising computational efficiency and reliability. Furthermore, for a subtle motion task in a narrow space, it is favorable for the algorithm to be able to find a correct answer with a desired resolution if it is recursively executed.

## IV. IMPLEMENTATION ON A HUMANOID ROBOT

The Honda humanoid testbed is a walking robot with 26 degrees of freedom. Since it is a very challenging problem to instantly update a manipulation task for such a complex robot, we have tested the proposed system on the robot hardware to evaluate effectiveness of the approach. The task objective in the experiment is to repeat movements between a start point and a goal point with the right hand. The goal point changes the position at each iteration. The robot should avoid every moving obstacle while moving between the two points. Furthermore, we have tested if the motion is robust when an undetected collision occurs.

### A. Elastic Strip Framework

As explained in the previous section, maintaining collision checking efficiency and reliability in spite of an increased complexity of the robot structure and the manipulation task motion is important for successful implementations of elastic strips on high d.o.f. robots in dynamic environments. We choose to use the *Adaptive Dynamic Collision Checking Algorithm* [26] when implementing the elastic strip on the Honda humanoid testbed. The algorithm is capable of detecting collisions on continuous motions without being significantly affected by geometric and kinematic complexities. When two robot postures and the continuous joint angle trajectory between the two robots are provided, the continuous motion is guaranteed to be collision free if the following condition is satisfied.

$$D_{travel} < d_1 + d_2 \tag{16}$$

where $D_{travel}$ is the longest distance of a body point when the robot travels along the trajectory and $d_1$, $d_2$ are distances to the closest obstacles from the two robot postures at both ends. For a point $p$ of the robot body, let $C_p(q_1, q_2, D)$ be the union of every curve connecting the body point $p$ of configurations $q_1$ and $q_2$ with a length less than $D$. We can also define a volume $V_{travel}(q_1, q_2, D)$ as the union of $C_p(q_1, q_2, D)$ of all points on the given robot body. When the longest travel distance $D_{travel}$ is computed, $V_{\mathscr{P}_c}$, a volume swept by the actual motion, is included in $V_{travel}(q_1, q_2, D_{travel})$. The volume can not have intersect with obstacles if the collision free condition (16) is satisfied. Thus, $V_{travel}(q_1, q_2, D_{travel})$ is chosen as an elastic tunnel which satisfies both conditions (2) and (3). The strategy in this method is to select first an elastic tunnel which is guaranteed to include $V_{\mathscr{P}_c}$ and check then if the tunnel

has any intersection with obstacles by checking (16). This approach is very efficient since the elastic tunnel is implicitly constructed by computing $D_{travel}$. The computational load in finding $D_{travel}$ does not increase significantly even if the workspace trajectory, kinematic structure, or robot geometries become highly complex, which make the collision checking algorithm suitable to manipulation tasks of high d.o.f. robots. Furthermore, since $D_{travel}$ decreases as the trajectory segment is divided into shorter pieces, collision checking solutions can be found with any desired resolution by recursively executing the algorithm. This type of collision checking algorithm does not make a simplifying assumption on the robot motion when checking if the robot motion along the trajectory is completely included in the elastic tunnel. Thus it does not fail to detect any collision along the continuous motion.

In computing $D_{travel}$ by the method explained in [26], the articulated robot model is rooted at the humanoid robot's hip position and the travel distance of the hip point is added to $D_{travel}$. If any segment is found not to satisfy (16), a new robot configuration is created in the middle of the invalid segment to produce a local collision avoidance motion at the narrow environment and to provide a smaller travel distance between configurations. Using these approaches, the free space motion search can refine the resolution. To find a redundant node along the elastic strip, it checks for collisions of the path between the $i^{\text{th}}$ node and the $(i+2)^{\text{th}}$ node. If the collision free condition $D_{travel(i,i+2)} < d_i + d_{i+2}$ holds, the $(i+1)^{\text{th}}$ node is found to be redundant and removed.

To stop the recursive execution at a desired resolution, a segment is considered as collision-free if two configurations are very similar to each other and the risk of collision at each discrete configuration is small enough, which means all of the following conditions are met.

$$\begin{array}{l} \Delta(q_1 - q_2) < \epsilon_1 \\ D_{travel} - (d_1 + d_2) < \epsilon_2 \\ d_1 > 0 \ , \quad d_2 > 0 \end{array} \tag{17}$$

If none of the conditions (16) and (17) are met, the path tracking should be paused and a new request should be sent to the global motion planner to provide a new solution, which resets the elastic strip.

A robot model at each node is controlled by the torque as

$$\begin{array}{l} \Gamma = \Gamma_{col.avoid.} \\ +{N_{col.avoid.}}^T(\Gamma_{hand\ pos.} + {N_{hand\ pos.}}^T \Gamma_{posture}) \end{array} \tag{18}$$

where hands are controlled to stay at intermediate positions in the work space by

$$\Gamma_{hand\ pos.} = {J_{hand\ pos.}}^T F_{hand\ pos.} \tag{19}$$

$J_{hand\ pos.}$ is 3 dimensional position Jacobian in Cartesian space and $F_{hand\ pos.}$ is a force to pull the hand position to the desired point. The desired point is set by interpolating desired hand positions at neighbouring nodes. $\Gamma_{posture}$ is composed of forces for the balance control, the internal contraction force $F^{int}$, and the default joint angle control. $\Gamma_{col.avoid}$ is a push torque to avoid collision, defined as

$$\Gamma_{col.avoid.} = {J_p}^T F_p^{col.avoid.} \tag{20}$$

where $J_p$ is a position Jacobian at the closest point $p$. The collision avoidance force is defined by the range of the closest distance to obstacles, $d$, as

$$F_p^{col.avoid.} = \begin{cases} F_p^{ext} + \Lambda(q)(k_v \dot{x}) & \text{if } d < d_0 \\ \Lambda(q)(k_v \dot{x}) & \text{if } d_0 \le d < d_{damp} \\ 0 & \text{if } d_{damp} \le d \end{cases} \tag{21}$$

The corresponding null space matrix $N_{col.avoid.}$ is also defined by $d$ as

$$N_{col.avoid.} = \begin{cases} I - J_p^T \bar{J}_p^T & \text{if } d < d_{damp} \\ I & \text{if } d_{damp} \le d \end{cases} \tag{22}$$

For smooth transitions in motion, the avoidance motion is damped when $d < d_{damp}$ where $d_{damp}$ is set as $d_0 < d_{damp}$.

### B. Motion Interpolation

The motion interpolation module updates the robot control commands while scanning the elastic strip ahead with a robot model. In this hand position tracking task, a joint angle interpolation control produces the tracking torque vector as

$$\Gamma_{tracking\ task} = A(k_p(q - q_d(u)) + k_v \dot{q}) + B + G \tag{23}$$

where $A$ is the inertia matrix, $B$ is the centrifugal and Coriolis force vector, $G$ is the gravity vector. $q$ and $\dot{q}$ are the current joint position and velocity of the interpolation robot model. $k_p$ and $k_v$ are position and velocity gains, and $q_d(u)$ is a desired joint vector proceeded by $u$ as:

$$q_d(u) = q_1(1 - u) + q_2 u \tag{24}$$

At each update, $u$ increases by $\Delta u$ limited by a constant $\alpha$ as

$$\Delta u = \min\left( \frac{\alpha}{\max_j |q_{1,j} - q_{2,j}|} \ , \ \beta \right) \tag{25}$$

where $q_{i,j}$ is the $\text{j}^{\text{th}}$ joint angle of the $\text{i}^{\text{th}}$ node for $j = 1, \cdots, n$ and $n$ is the degree of freedom of the robot. $\alpha$ is the maximum joint angle change during a single iteration and $\beta$ sets the minimum number of iterations between two nodes. Here, $0 < \alpha$ and $0 < \beta < 1$.

### C. Robot Motion Control

*1) Lower Body Task Control:* The main task of the lower body control is to place the upper body at a desired position and orientation to support stable manipulation task motion. Thus, the lower body controller performs 6 d.o.f. position and orientation tracking of the hip link by the configuration output of the motion interpolation module. The orientation and acceleration of the torso is monitored by a gyro sensor, and the actual position and orientation is estimated by the measured joint angles of the lower body. The Honda ZMP(zero moment point) controller uses this information to maintain robust motion of the whole body.

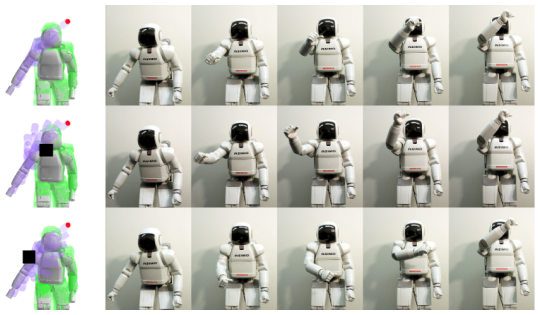

Fig. 3: A reactively modified motion plan to avoid virtual obstacles in the simulator and its tracking motion by a real robot. The default motion plan (1st row) is modifed to move the arm above the obstacle (2nd row) or below the obstacle (3rd row) while maintaining the safety distance. The real robot follows the commanded postures which is continuously generated by the interpolation module.

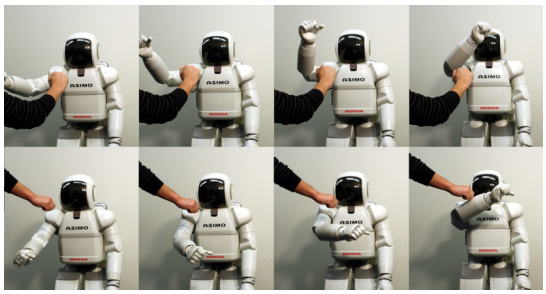

Fig. 4: Task motions of the Honda humanoid testbed while avoiding a human arm in the work area. The human body is monitored as an obstacle by a 3D depth sensor. The two motions result from the original task motion after being modified by the Elastic Strip framework.

| | Average (sec) | STD (sec) | Num. of Updates |
|---|---|---|---|
| Update Time | 0.0523 | 0.0006 | 1 |
| Reaction Time | 0.368 | 0.020 | 5.0 |
| Stabilization Time | 4.79 | 0.67 | 72.5 |

TABLE I: Time for updating the plan, reacting to moving obstacles, and stabilizing when obstacles stop moving. The last column shows the iteration numbers for reacting and stabilizing in average of 10 trials. The performance is measured in a system of Intel Core i7 CPU (8 cores of 2.8GHz) with Windows7 operating system.

*2) Upper Body Task Control:* The upper body of the robot has a controller to execute both arms' task motions which are updated by the interpolation module. Since the interval of the interpolation module iteration is irregular and less frequent than the interval of the hardware control loop, the desired torque command from the planning layer results in a rough output from the transformer. Here, the On-line Trajectory Generation (OTG) algorithm [27] computes a smooth trajectory command at each instance from the irregular velocity output of the transformer. The OTG algorithm is able to produce a smooth trajectory instantly, which connects any state and a desired goal state for given constraints with a synchronized trajectory in the shortest time possible.

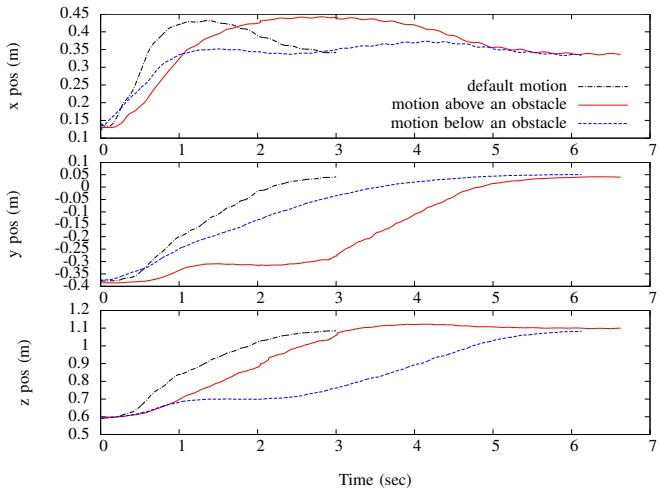


Fig. 5: Position trajectories of the right hand between the start position and the goal position. It shows that the collision avoidance motions take more time to reach the goal than the default motion.

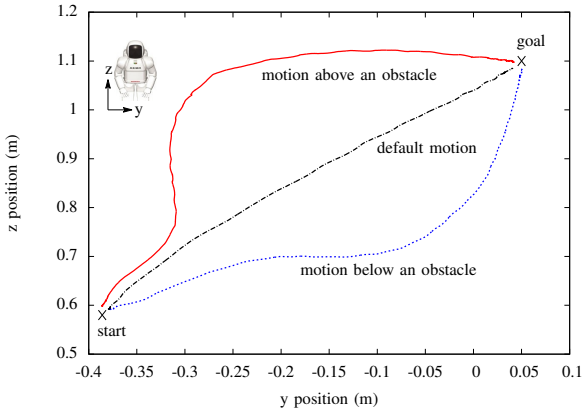


Fig. 6: Right hand trajectories recorded during real-time tests. They show how to reach a goal position by the default task motion and avoidance motions.

## V. EXPERIMENTAL RESULTS & DISCUSSION

The system performance of reacting to environment changes is evaluated by experiments. In a simulator a virtual obstacle is moved and the position is sent to the Elastic Strip module. The time for updating the motion plan, reacting to moving obstacles, and stabilizing when obstacles stop moving is measured as in Table I for different obstacle motions. Figure 3 shows the default task motion and modified motions to avoid an obstacle. Modified arm motions are generated from the default state in real-time Each motion is still able to reach the goal without any collision. Table I shows that the elastic strip is updated at 20Hz speed, and it reacts to moving obstacles in about 0.3 sec. or in 5 iterations which is measured between the moment when an obstacle comes closer than a safety distance and the moment when the arm reaches 0.1 m/s speed in the avoiding direction. The whole elastic strip motions stabilize in less than 5 seconds or in around 70 iterations after obstacles stop moving. Figure 4 shows real-time experiments using a vision sensor to detect

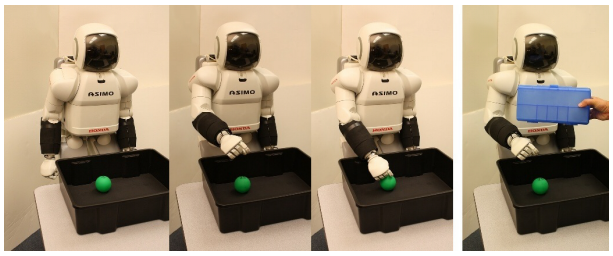

Fig. 7: A task motion in a more complex environment. The robot moves the right hand to reach the green ball in the tray. During the task motion, the robot has to avoid collisions with the table and the tray as well as the moving obstacle (the blue box). The position of the blue box is monitored by a vision camera and sent to the Elastic Strip framework in order to update the environment information.

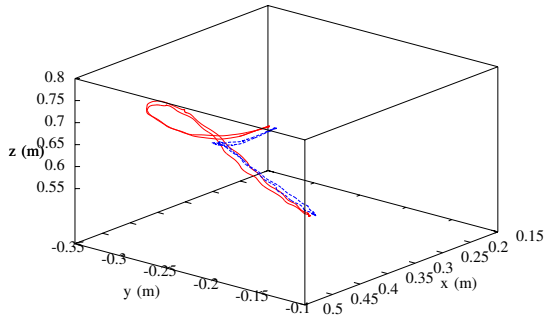


Fig. 8: Position paths of the right hand during one cycle of the task motion. The red line is for the default task motion and the blue line is for the motion modified in order to avoid collision. It shows that the motion is more constrained in the z direction when there is a moving obstacle.

obstacle movements. In the test, the robot follows a different motion paths in order to avoid collisions with the human and to reach the manipulation task goal position. Figure 6 shows right hand trajectories reactively generated under three different situations.

Figure 7 shows a task motion in a more complex situation. The robot moves right hand into the tray placed on the table to reach the green ball in the tray. The arm needs to avoid other moving objects while reaching the ball. The default task motion without a moving obstacle is compared with a motion which is reactively modified to avoid a collision. Figure 8 shows the right hand position trajectories of the default motion and the modified motion during one cycle of the task motion. The modified motion is more constrained in z direction since the obstacle stays above the arm motion and produces virtual forces to push the path down. Figure 9 shows more clearly how much the hand motions are modified and constrained in each direction. The motion does not change much in x and y directions. However, the z direction motion is significantly reduced by the moving obstacle. In the test, a collision-free path is successfully maintained and the robot can reach the same goal position within a similar time. Figure 10 shows that the joints of the shoulder (joint

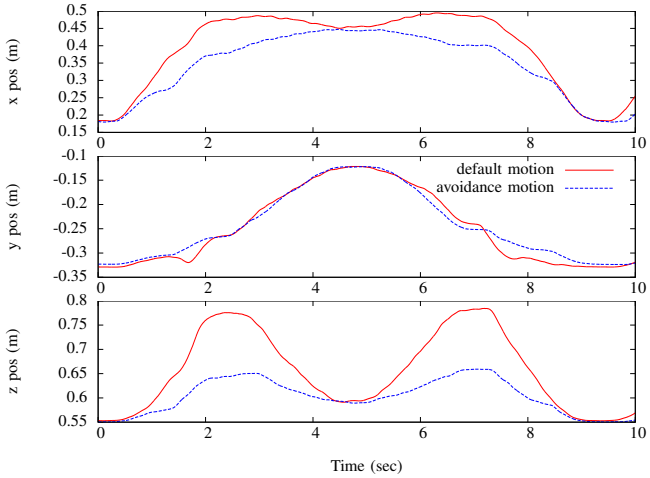


Fig. 9: Position trajectories of the right hand during one cycle of the task motion. It shows that the motions in the x and y directions are not changed much in this example. The path is modified mostly in the z direction by the moving obstacle.

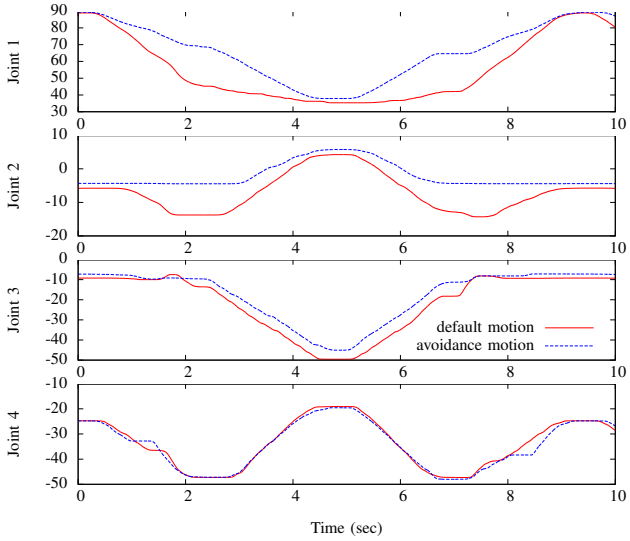


Fig. 10: Joint angle trajectories of the right arm during one cycle of the task motion. The joint 1 and 2 are the two joints of the shoulder, and the joint 3 and 4 are the tow joints of the elbow. They show that the shoulder motion is changed more than the elbow motion.

1 and 2) are more involved in the motion change for the collision avoidance in the example. In contrast, the elbow joints (joint 3 and 4) move with a very similar trajectory in the both cases.

The motions under exceptional situations are also tested for the events when colliding with undetected obstacles while executing tasks in real time. In the robot test, an obstacle is placed on the elastic strip path and the vision sensor is disabled to simulate the failure of obstacle detection. As seen on the Figure 11, even when the moving arm is blocked by sudden unforeseen contacts, the elastic strip stably stops advancing to the goal position and pauses until the robot resumes the desired motion. This experiment shows the effectiveness of combining compliant force control and the plan modification approach in real time.

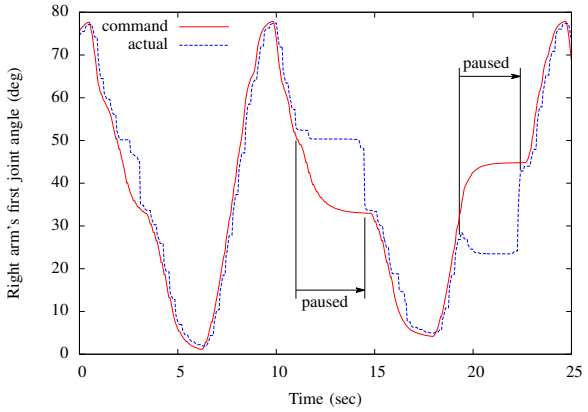


Fig. 11: Joint angle trajectory of the right arm's first joint under exceptional situation. The trajectory until 10 second shows normal tracking motion. At 11 and 19 seconds, collisions undetected by sensors prevent the robot from following the elastic strip command. When the actual motion cannot follow the command, the elastic strip stops advancing to the goal position and pauses until the robot resumes the commanded motion.

## VI. CONCLUSIONS

A reactive motion modification approach based on the Elastic Strips framework has been presented for real-time manipulation tasks of humanoid robots working in unpredictably changing environments. Real-world experimental results on a humanoid robot with 26 degrees of freedom were reported and discussed. It was shown that the proposed system adapts the robot's motion paths effectively when sudden changes in the environment are detected. The system was shown to be robust by handling exceptional events such as undetected collisions to the robot. This type of real-time motion generation can take into account kinematic and dynamic constraints and react to sudden environmental changes. Thus, it will play a key role for future implementations of the multiple tasks of highly redundant robots. In the next step, we will expand the online motion planning system, such that also in-contact motions during physical human-robot interaction tasks can be taken into account.

## REFERENCES

[1] Y. Koga and J.-C. Latombe, "On multi-arm manipulation planning," in *International Conference on Robotics and Automation*, 1994, pp. 945–952.

[2] J.-M. Ahuactizin, K. Gupta, and E. Mazer, "Manipulation planning for redundant robots: A practical approach," *International Journal of Robotics Research*, vol. 17, no. 7, pp. 731–747, Jul. 1998.

[3] T.-Y. Li and J.-C. Latombe, "On-line manipuation planning for two robot arms in a dynamic environment," *International Journal of Robotics Research*, vol. 16, no. 2, pp. 144–167, Apr. 1997.

[4] T. Simeon, J.-P. Laumond, J. Cortes, and A. Sahbani, "Manipulation planning with probabilistic roadmaps," *International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 729–746, Jul. 2004.

[5] N. Amato and Y. Wu, "A randomized roadmap method for path and manipulation planning," in *International Conference on Robotics and Automation*, 1996, pp. 113–120.

[6] M. Kallmann, A. Aubel, T. Abaci, and D. Thalmann, "Planning collision-free reaching motions for interactive object manipulation and grasping," *EUROGRAPHICS*, vol. 22, no. 3, pp. 313–322, 2003.

[7] J. Kuffner and J.-C. Latombe, "Interactive manipulation planning for animated characters," in *The Eighth Picific Conference on Computer Graphics and Applications*, 2000, pp. 417–418.

[8] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Motion planning for humanoid robots," *Springer Tracts in Advanced Robotics*, vol. 15, pp. 365–374, 2005.

[9] A. Dietrich, T. Wimböck, and A. Albu-Schäffer, "Dynamic whole-body mobile manipulation with a torque controlled humanoid robot via impedance control laws," in *International Conference on Intelligent Robots and Systems*, 2011, pp. 3199–3206.

[10] A. Dietrich, T. Wimböck, H. Taubig, A. Albu-Schäffer, and G. Hirzinger, "Extension to reactive self-collision avoidance for torque and position controlled humanoids," in *International Conference on Robotics and Automation*, 2011, pp. 3455–3462.

[11] K. Nagasaka, Y. Kawanami, S. Shimizu, T. Kito, T. Tsuboi, A. Miyamoto, T. Fukushima, and H. Shimomura, "Whole-body cooperative force control for a two-armed and two-wheeled mobile robot using generalized inverse dynamics and idealized joints," in *International Conference on Robotics and Automation*, 2010, pp. 3377–3383.

[12] A. De Luca and F. Flacco, "Integrated control for phri: Collision avoidance, detection, reaction and collaboration," in *International Conference on Biomedical Robotics and Biomechatronics*, 2012.

[13] A. De Luca, F. Flacco, and O. Khatib, "Motion control of redundant robots under joint constraints: Saturation in the null space," in *International Conference on Robotics and Automation*, 2012, pp. 285–292.

[14] H. Iwata, H. Hoshino, T. Morita, and S. Sugano, "Human-humanoid physical interaction realizing force following and task fulfillment," in *International Conference on Intelligent Robots and Systems*, 2000, pp. 522–527.

[15] M. Behnisch, R. Haschke, and M. Gienger, "Task space motion planning using reactive control," in *International Conference on Intelligent Robots and Systems*, 2010, pp. 5934–5940.

[16] H. Sugiura, M. Gienger, H. Janssen, and C. Goerick, "Real-time collision avoidance with whole body motion control for humanoid robots," in *International Conference on Intelligent Robots and Systems*, 2007, pp. 2053–2058.

[17] D. Berenson, S. Srinivasa, and J. Kuffner, "Task space regions: A framework for pose-constrained manipulation planning," *International Journal of Robotics Research*, vol. 30, no. 12, pp. 1435–1460, Oct. 2011.

[18] S. Quinlan and O. Khatib, "Elastic bands: connecting path planning and control," in *International Conference on Robotics and Automation*, 1993, pp. 802–807.

[19] O. Brock and O. Khatib, "Elastic strips: A framework for motion generation in human environments," *International Journal of Robotics Research*, vol. 21, no. 12, pp. 1031–1052, Dec. 2002.

[20] Y. Yang and O. Brock, "Elastic roadmaps - motion generation for autonomous mobile manipulation," *Autonomous Robots*, vol. 28, pp. 113–130, 2010.

[21] O. Khatib, L. Sentis, and J. Park, "A unified framework for whole-body humanoid robot control with multiple constraints and contacts," in *European Robotics Symposium 2008*, ser. Springer Tracts in Advanced Robotics, H. Bruyninckx, L. Přeučil, and M. Kulich, Eds. Springer, Berlin, Heidelberg, Germany, 2009, vol. 44, pp. 303–312.

[22] O. Khatib, K. Yokoi, K. Chang, D. C. Ruspini, R. Holmberg, and A. Casal, "Coordination and decentralized cooperation of multiple mobile manipulators," *Journal of Robotic Systems*, vol. 13, no. 11, pp. 755–764, 1996.

[23] O. Khatib, "Inertial properties in robotics manipulation: An object-level framework," *The International Journal of Robotics Research*, vol. 14, no. 1, pp. 19–36, February 1995.

[24] O. Khatib, P. Thaulad, T. Yoshikawa, and J. Park, "Torque-position transformer for task control of position controlled robots," in *International Conference on Robotics and Automation*, 2008, pp. 1729–1734.

[25] T. Yoshikawa and O. Khatib, "Compliant humanoid robot control by the torque transformer," in *International Conference on Intelligent Robots and Systems*, 2009, pp. 3011–3018.

[26] F. Schwarzer, M. Saha, and J.-C. Latombe, "Adaptive dynamic collision checking for single and multiple articulated robots in complex environments," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 338–353, Jun. 2005.

[27] T. Kröger and F. M. Wahl, "On-line trajectory generation: Basic concepts for instantaneous reactions to unforeseen events," *IEEE Trans. on Robotics*, vol. 26, no. 1, pp. 94–111, Feb. 2010.