Optimal Whole Body Motion Planning of Humanoid with Articulated Spine for Object Manipulation in Double Support Phase

Hari Teja K
International Institute Of Information Technology Hyderabad
Telangana, India
hari.teja@research.iiit.ac.in

Francis James
International Institute Of Information Technology Hyderabad
Telangana, India
francisjames10@gmail.com

Suril V. Shah
International Institute Of Information Technology Hyderabad
Telangana, India
surilshah@iiit.ac.in

ABSTRACT

In this paper we consider the motion planning problem of humanoid with articulated-spine for object manipulation in double support phase. Complexity of motion planning increases due to higher degrees-of-freedom (DOF), redundancy arising due to articulated spine, inherent underactuation and stability constraint. Additionally, loop-closure constraints arise during double-support phase, and self-collision constraints exist independent of the task to be performed. These problems make motion planning of humanoids a challenging task. In this work, we address the above issues by proposing a sampling based approach for planning the motion of the humanoid. Our approach to the problem is based on RRT* which can generate asymptotically optimal paths. The proposed approach deals with the stability constraints by rejective sampling approach which divides random configurations generated into valid and invalid configurations. Additionally, the loop-closure constraints are tackled by separating the humanoid into two open-kinematic sub-chains, and then generating random configurations in one sub-chain whereas the remaining sub-chain uses inverse kinematics for closure. Efficacy of the proposed approach is demonstrated for whole-body motion planning of a 25-DOF humanoid in generating asymptotically optimal end-effector paths.

Keywords

Articulated spine, optimal paths,RRT*,redundant manipulator,high Degrees of freedom, static-stability, loop-closure, motion planning, object manipulation, rejective sampling, stability constraints

1. INTRODUCTION

Humanoid robots are very well suited for manipulation of objects in complex environments due to their highly redundant structure. Their whole-body motion planning in double support phase can be challenging due to high degrees-of-freedom (DOF), loop-closure, static stability and other task constraints. This paper proposes the use of sampling based asymptotically optimal path-planning method for object manipulation by humanoid in double support phase. As humanoid is a redundant system,it can be utilized to specify multiple tasks or optimize certain states during a task. Much work has been done in tackling this problem using Jacobian methods [1] [2] dealing with redundant manipulators in general.

AIR '15, July 02 - 04, 2015, Goa, India

DOI: http://dx.doi.org/10.1145/2783449.2783479

1.1 RELATED WORK AND CONTRIBUTION

This section reviews related work, and concludes with the contributions of this work for the motion planning of humanoid.

1.1.1 Randomized approach for motion planning

The application of randomized methods for path planning in high dimensional configuration spaces has been widely studied [3][4]. Rapidly Exploring Random search Trees (RRT) [5] have found a wide spread application due to their generalized approach in solving problems including differential constraints and configuration spaces with high dimensions. RRT achieves rapid exploration by incrementally sampling random states in state space and expanding the tree towards the random states. Many variants of RRT such as E-RRT, RRT-connect and T-RRT [6] [7] can be found in literature, each of which improves upon the basic RRT algorithm for real-time applications or cost inclusion. These have been applied across areas of robotics, molecular biology and animation. RRT* [8] algorithm is another variant of RRT that is proven to lead to asymptotically optimal paths as number of iterations increases. Attempts have been made to use RRT like approaches for whole body task planning of humanoids[10] along with local Inverse Kinematic techniques as an approach to integrate collision avoidance. Recently in [11], RRT-connect algorithm along with an Inverse Kinematic solver has been used for Whole-body motion planning of humanoid for manipulation of objects. Although time performance of planning algorithm has been improved, it suffers from optimality issues due to the inherent nature of the RRT algorithm used, which has almost zero probability of finding an optimal path.

RRT* algorithm overcomes this drawback and its performance in high dimensional configuration spaces has been proved to be effective in [9] where RRT* was applied for

dual-arm motion planning of PR2 manipulator. The use of RRT* for whole-body motion planning of a humanoid robot with articulated spine in double support phase is not a trivial extension of [8], and poses some obvious challenges as mentioned earlier. Such work has not been reported in literature to the best of the authors' knowledge, and is taken up in this paper.

1.1.2 Loop-Closure Constraints

Mechanisms involving closed kinematic chains can be solved as an Inverse Kinematic (IK) problem which yields a set of non-linear equations. A variety of methods to solve these equations have been proposed ranging from Jacobian-based methods to Polynomial continuation [15]. These iterative methods are difficult to apply along with probabilistic planning techniques because of the possibility of multiple solutions and a lack of generality as the loop-closure equations are dependent on the system under study. For a randomized sampling based methodology which requires exact solutions, these methods bring in additional complexities .

The problem is approached in [13] by breaking the kinematic chain into set of open sub-chains, applying standard probability road map (PRM) random sampling techniques and forward kinematics to one set of sub-chains and inverse kinematics to remaining set to enforce loop-closure. This has proven to be particularly useful in preserving the randomized sampling philosophy and generating an accurate closed form solution. This method has been further improved in Random Loop Generator (RLG) algorithm [14] in terms of time required by limiting the generation of random configurations to a set of configurations which fall in the approximated region of closure satisfaction called reachability space. This algorithm is more useful when dealing with large kinematic chains where the probability of generating configurations satisfying loop-closure constraints becomes very low.

1.1.3 Contribution

Our contribution is to the problem of generating asymptotically optimal paths with statically stable configurations of humanoid for the manipulation of objects using RRT*. These generated paths can then be converted into dynamically stable trajectories in a seperate stage independent of path-generation stage, and is not considered in this paper. Moreover, we apply methodology of [13] and exploit the relatively simple nature of closed kinematic chain formed by lower body during double-support phase with only six joints (two 3-DOF joints,two 2-DOF joints, and two joints with 1-DOF), which can be divided into two sets of three joints each. Next, forward kinematics is applied for generating random configurations of the upper body and one set of the lower body kinematic chain, and using Inverse kinematics on the remaining set for loop-closure.

1.1.4 Paper Layout

Next section presents the model of the humanoid considered in this paper. Section 3 presents the Whole Body Motion Planning of biped, which is further divided into subsections covering preliminaries for the algorithm and the method followed in this paper for completing the task. Section 4 presents the experiment and results followed by section 5 which concludes this paper.

2. HUMANOID CONFIGURATION

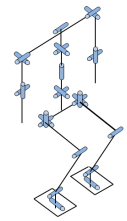


Figure 1: Humanoid(Biped) Diagram

The humanoid model used for our motion planning purpose is inspired from *INRIA's POPPY* [16] robot with additional 2-DOF's included in feet of the robot for improved reachability. A total number of 26 joints are present, out of which joint at the neck is available for head Positioning which is not considered in present paper. Hence, the humanoid as an open-kinematic chain has 25-DOF's. Each ankle has a 2 DOF universal joint, each knee has a 1 DOF revolute joint, hip has two 3 DOF spherical joints on both sides and Torso has 5-DOF's. Moreover, each shoulder and elbow has a 2 DOF universal joint. In the double-support phase, applying Grubler's equation for mobility M with L number of links , J_i number of i DOF joints[1] to the closed chain formed by lower body, it is found that 12 Joint Variables in the lower body account for only 6-DOF,Hence the humanoid remains with a total of 19-DOF.

3. WHOLE BODY MOTION PLANNING

This section deals with the method followed for motion planning of humanoid in double-support phase. First the problem statement is defined, then preliminaries of motion planning task are given and finally the application of these concepts for asymptotically optimal motion planning of humanoid is demonstrated.

3.1 Problem Statement

Let $X \subseteq \mathbb{R}^n$ be the set containing all joint parameters of our humanoid. Consider X_{rl}, X_{ll}, X_{ub}, X_{lb} as sets of joint variables for the right-leg region, left-leg region, upper body region and lower body region, respectively. Note that $X_{lb} := X_{rl} \cup X_{ll}$, $X := X_{ub} \cup X_{lb}$. We know that in double-support phase the degrees of freedoms of the humanoid are reduced. Hence, if X_{cs} is the configuration space of humanoid in the double-support phase then $X_{cs} \subset X$.

Consider X_{ind}, X_{dep} as the sets containing independent and dependent joint variables, respectively, of the lower body. Since the DOF of closed-loop of lower body becomes six in double support phase by Gruebler's Equation, we safely consider six right-leg joint variables as independent variables and remaining six joint variables of left-leg as dependent variables, therefore we choose $X_{rl} := X_{ind}$ and $X_{ll} := X_{dep}$. Hence, mapping $M : X_{ind} \rightarrow X_{dep}$ can be written as $M : X_{rl} \rightarrow X_{ll}$ which can indeed be expressed as composition of two mappings $'F'$ and $'I'$, where $F : X_{rl} \rightarrow \mathbb{R}^3$ is the forward kinematics mapping from right-leg to the three dimensional cartesian space hip coordinates, and $I : \mathbb{R}^3 \rightarrow X_{ll}$ is the inverse kinematics mapping from hip-coordinates to joint space of left-leg. Hence, the problem of loop closure

[1] $M = 6(L-1) - 5J_1 - 4J_2 - 3J_3 - 2J_2 - J1$

can be divided into two simple sub problems. Note that such inverse and forward kinematics mapping are well studied in robotics literature [17], and hence, not reported here for brevity.

It is worth noting that not all $X_{cs} \subset X$ are attainable due imposed stability , collision and joint limit constraints. If the constraint region is denoted by X_{cons}, then $X_{free} := X_{cs}/X_{cons}$. Given the initial configuration (x_{init}) and goal configuration (x_{goal}), our objective is to find path $\Omega : [0,1] \rightarrow X_{free}$, where $\Omega(0) = x_{init}$ and $\Omega(1) = x_{goal}$, such that path cost is minimum and mapping $M := \mathrm{I} \circ \mathrm{F}$.

3.2 ALGORITHMS AND METHODS

For generating asymptotically optimal path for end-effector of humanoid we use RRT*. RRT* was first introduced in [8]. It is an incremental sampling based algorithm that performs path rewiring in each iteration of node generation as an optimization step. In this section the generic RRT* algorithm as proposed in [8] is presented in our notation for better understanding of the proposed implementation, then it is extended for motion planning of humanoid.

3.2.1 RRT* Algorithm preliminaries

Working of RRT* is similar to RRT. It achieves rapid exploration by incrementally sampling random configuration in configuration space and expanding the tree towards the random configuration as shown in Fig. 2. Additionally, optimization of path produced is performed at each iteration by rewiring the edges of tree in order to produce low-cost paths, as illustrated in Fig. 3.

Implementation of RRT* is given in Algorithm 1. Algorithm 1 uses the function *Extend*()(Algorithm 2) to expand the tree. *Extend*() performs rewiring as shown in Algorithm 3, which uses the functions *BestParent*()(Algorithm 4) and *BestChild*()(Algorithm 5) to find optimal path. Several other functions used while implementing RRT* are explained below:

RandGen() : This function returns random samples x_r from configuration space.

NearestNode() : Given a set V and any random sample $x_r \in$ X, *NearestNode*(V,x_r) returns the closest configuration x_{cls} in V to x_r in terms of the L2 norm.

NewNode() : Given a random sample x_r and output of *NearestNode*(V,x_r), *NewNode*(x_r,x_{cls}) generates new configuration from x_{cls} towards x_r.

NearNodes() : Given a set V and any configuration $x \in X$, *NearNodes*(V,x) gives the set of configurations in V that are near to x. Alternatively, let n be the number of configurations in V, then *NearNodes*(V,x) is the set of all configurations in V that lie inside a ball of volume $\mathcal{O}((logn)/n)$ [9] centered at x.

GoalPosReached() : Given set of vertices V and a target goal position $x_g \in X$, *GoalPosReached*(V,x_g) returns flag as 1 and the nodes n_f falling within acceptable limit of reaching goal position. While this is a generalised form of representation, for path-planning in configuration spaces *GoalPosReached*() uses forward kinematics equations of given robot to determine the position of end effector in each iteration.

Cost() : Given a set V and any configuration $x \in V$, *Cost*(V,x) gives the total defined cost to travel from x_{init} to x.

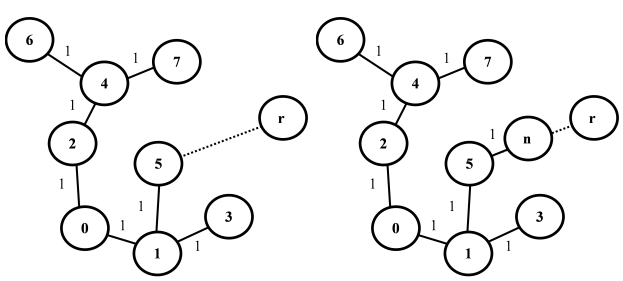
BestParent() : Given any configuration $x \in V$ and set of nearest nodes N returned from *NearNodes*(V,x) to the particular x, *BestParent*(N,x) returns the rewired tree in which the cost to travel from x_{init} to x is minimum.

BestChild() : Given any configuration $x \in V$ and set of nearest nodes N returned from *NearNodes*(V,x) to the particular x, *BestChild*(N,x) returns the rewired tree in which the cost to travel from x_{init} to each near node is minimum.

CollisonCheck() : Given a configuration $x \in X$ and total information of environment as a global variable matrix *map*, *CollisionCheck*(*map*,x) returns 1 if x lies in obstacle space X_{obs} of the map, else it returns 0.

BackTrack() : Given a configuration set V and an array of numbers n indexing elements in V, *BackTrack*(V,n) returns the paths formed between each node indexed by n and initial node (with node index 0).

The pseudocode for RRT* is given in Algorithms 1-5 in which V and E are considered as set of vertices and set of edges of tree T respectively, and *path* is the variable containing vertices of path generated from x_{init} to x_{goal}.



(a) (b)

Figure 2: Tree expansion in RRT* algorithm. a) A random point $'r'$ is generated in state space. b) Nearest node(node $'5'$ above) is expanded into state $'n'$ towards $'r'$.

Algorithm 1 void *main*(*map*, x_{init}, x_{goal})

```
V ← V.add(x_init)
E ← φ, T ← (V, E)
path ← φ
for i = number of iterations do
    x_r ← RandGen()
    [V, E] ← Extend(V, E, x_r)
    [flag, n_f] ← GoalPosReached(V, x_goal);
if flag=1 then
    path ← BackTrack(V, n_f)
else
    printf("path not found")
return path
```

Algorithm 2 *Extend*(V, E, x_r)

```
x_near ← NearestNode(V, x_r)
x_new ← NewNode(x_near, x_r)
collision ← CollisionCheck(x_new, map)
if collision ≠ 1 then
    V = V.add(x_new)
    E = E.add(x_near, x_new)
    [V, E] ← Rewire(x_new, V, E)
return (V, E)
```

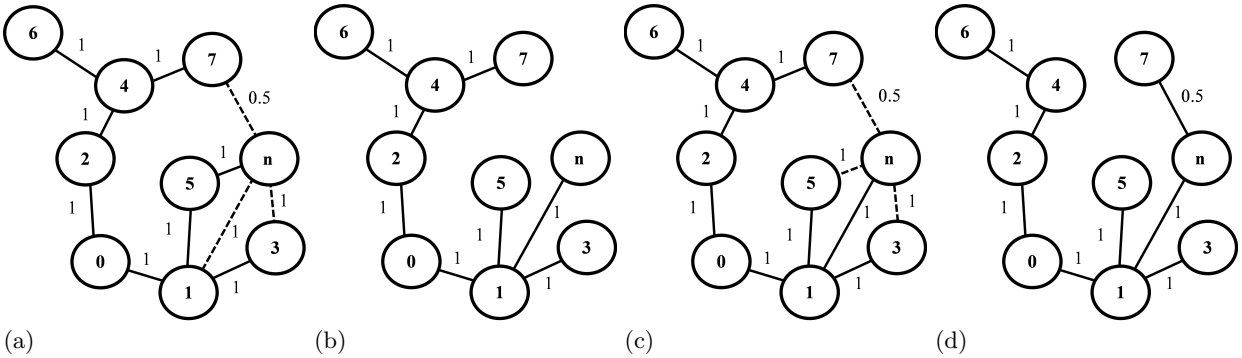


Figure 3: Rewiring step. a) Near vertices of new node $'n'$ are connected (using dashed line) as parent vertices for $'n'$. b) Path with lowest cost from initial vertex $'0'$ to reach $'n'$ is found and the respective low cost nearest vertex is assigned as a best parent vertex to $'n'$, remaining edges are deleted. c) Remaining nearest vertices of new node $'n'$ are connected as children (using dashed line) to $'n'$. Path cost from $'0'$ towards each near vertex of $'n'$ with $'n'$ as parent is compared with the original path in (b), d) Edges corresponding to high cost paths are deleted and low-cost paths remain.

Algorithm 3 $Rewire(x_{new}, V, E)$

$neighbours \leftarrow NearNodes(x_{new}, V, E)$
if $size(neighbours) > 0$ **then**
  $[V, E] \leftarrow BestParent(x_{new}, neighbours, V, E)$
  $[V, E] \leftarrow BestChild(x_{new}, neighbours, V, E)$
else return (V, E)
return (V, E)

Algorithm 4 $BestParent(x_{new}, neighbours, V, E)$

for each integer i in $length(neighbours)$ **do**
  $E.remove(:, x_{new})$
  $E.add(neighbour(i), x_{new})$
  $Pcost(i) \leftarrow Cost(V, E, x_{new})$
$BestParent_{index} \leftarrow min_{index}(Pcost)$
$BestParent \leftarrow neighbours(BestParent_{index})$
$E.remove(:, x_{new})$
$E.add(BestParent, x_{new})$
return (V, E)

Algorithm 5 $BestChild(x_{new}, neighbours, V, E)$

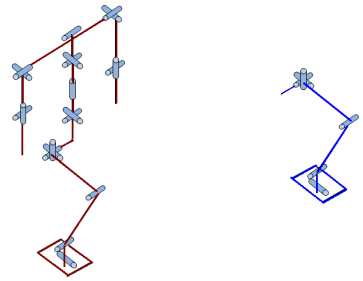
$E_{varied} \leftarrow E$
for each integer i in $length(neighbours)$ **do**
  if $i \neq BestParent_{index}$ **then**
    $E_{varied}.remove(:, neighbour(i))$
    $E_{varied}.add(x_{new}, neighbour(i))$
    $Pcost(i) \leftarrow Cost(V, E, x_{new})$
    $Pcost_{varied}(i) \leftarrow Cost(V, E_{varied}, x_{new})$
    if $Pcost(i) > Pcost_{varied}(i)$ **then**
      $E.remove(:, neighbour(i))$
      $E.add(x_{new}, neighbour(i))$
return (V, E)

3.2.2 Humanoid Motion Planning

The kinematic diagram of humanoid is shown in Figure.1. The total number of joints are 25, but in the double-support phase the DOF of this humanoid are limited to 19 (13-DOF Upper-Body, 6-DOF Lower-Body).

To apply RRT* on humanoid, $RandGen()$ function is modified to generate configurations pertaining to joint limits of the humanoid. Due to the dependency of joint variables in the lower-body, RRT* cannot be directly applied taking set of all joint variables as input configuration parameters for the algorithm. So the whole kinematic chain of body is split into two sub-chains (A) and (B), where (A) as shown in Fig.4(a) includes right leg, hip, upper body and hands (right, left), and *(B)* which is presented in Fig.4(b) is the 6-DOF left leg.

Forward kinematic mapping $F : X_{rl} \rightarrow \mathbb{R}^3$ from right toe to hip is estimated using transformations based on DH parameters with ground as the global reference frame having origin at the right toe. Inverse kinematic mapping $I : \mathbb{R}^3 \rightarrow X_{ll}$ has been formulated for the left leg in closed form by



(a) Kinematic sub-chain (A) (b) Kinematic sub-chain (B)

Figure 4

decoupling method. The RRT* algorithm is applied considering only (A) kinematic chain configurations, and at every iteration inverse kinematics mapping *'I'* for left leg is calculated for given *'F'* obtained from forward kinematics, and checked for the closure of right-leg and left-leg. The configurations that comply with these loop-closure constraints are treated as valid nodes and tree is updated with them.

3.2.3 Constraint Handling and Cost

Constraints handling is the main cause of high time requirements of the humanoid motion planning. Since, in this paper we are not addressing the time performance of the

motion planning which is already done in [11], these constraints are not used in creating manifolds to reduce the span of random configurations. Joint limits are the only intervals to which random configurations of humanoid are limited to. Constraints are checked for every iteration within motion planning algorithm itself for the validity of random configurations generated, and invalid configurations are not added to the tree. In this paper, we focus on the length of the path covered by end-effector as *Cost* function while others such as energy or control-effort can also be used.

To apply RRT* for humanoid, *Extend*() is modified into *BipedExtend*() with additional functions as given in Algorithms 6 and 7.

Algorithm 6 $BipedExtend(V, E, x_r)$

$x_{near} \leftarrow NearestNode(V, x_r)$
$x_{new} \leftarrow NewNode(x_{near}, x_r)$
$[x_{ll}, ClosureStatus] \leftarrow CheckClosure(x_{new}, BipedProps)$
$StabilityOK \leftarrow COMCheck(x_{near}, BipedProps, x_{ll})$
$collision \leftarrow CollisionCheck(x_{new}, x_{ll}, map)$
if $collision = 0$ and $ClosureStatus = 'OK'$ **then**
  if $StabilityOK$ **then**
    $V = V.add(x_{new})$
    $E = E.add(x_{near}, x_{new})$
    $[V, E] \leftarrow Rewire(x_{new}, V, E)$
return (V, E)

Algorithm 7 $CheckClosure(x_{new}, BipedProps)$

$Hip_{pos} \leftarrow ForwardKin(x_{new}, BipedProps)$
$x_{ll} \leftarrow InverseKin(Hip_{pos}, BipedProps)$
if x_{ll} *is valid* **then**
  $ClosureStatus =' OK'$
  return x_{ll},$ClosureStatus$
else
  $ClosureStatus =' NOT\ OK'$
  return $null$,$ClosureStatus$

The additional functions used in Algorithms 6 and 7 are explained as folows :

COMcheck() : Given a humanoid configuration $x \in X$, which is the set of all joint variables, and humanoid link lengths(*BipedProps*), *COMcheck*(x,*BipedProps*) calculates the projection of centre of mass (COM) on plane of walking and returns 1 if COM lies within the support polygon else returns 0.

CheckColsure() : Given a new node x_{new} and link lengths of humanoid(*Bipedprops*) as given in Table.1, *CheckClosure*(x_{new},*BipedPRops*) returns 1 if closure constraint is satisfied at x_{new} else returns 0. To achieve this *CheckClosure* runs Forward-Kinematics (*ForwardKin*()) for right leg and upper body (Chain-A) to estimate hip position and orientation, and validates the reachability of inverse kinematics solution(*InverseKin*()) obtained for the left leg (Chain- B).

4. EXPERIMENTS

We present the results of optimized end-effector path length of the humanoid which is an important aspect of humanoid motion planning.

For the implementation of this method a 25-DOF humanoid with two 4-DOF hands, two 6-DOF legs, 5-DOF articulated spine is considered. Our experiment considers 19-DOF as we perform it in double-support phase. One important point to be considered here is, for carrying out motion planning in this paper, a single-tree RRT* algorithm without any biasing is used. This results in large time requirements. As already mentioned, the focus of this paper is mainly in obtaining optimal paths. Without a loss in generality, the task of placing a spherical ball on top of table which is at equal height as the humanoid is considered for experiment. This task involves relatively lesser body movements compared with tasks that involve bending. Moreover, choosing spherical ball eliminates end-effector constraints. Its performance is evaluated with increasing number of iterations (1500,2000,2500,3000,3500,4000) and the position of centre of mass during humanoid motion is plotted. In the experiment the spherical object's mass is assumed negligible compared to the mass of humanoid so that it doesn't effect the static equilibrium of humanoid.

4.1 Results

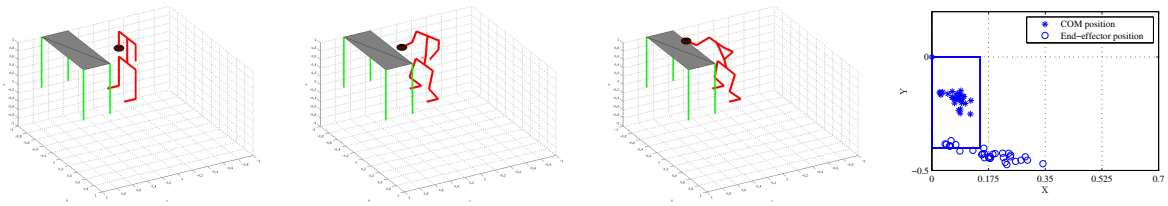
In this section we provide the simulation results to validate the effectiveness of the proposed methodology for humanoid motion planning.

Fig. 5 (a),(b),(c) shows the snapshots of the humanoid placing spherical object on table. It can be seen that it is successful in achieving the task without violating loop closure constraints, which is easily observable as the legs are not being separated at the Hip upon movement. In Fig. 5 (d), the blue polygon is the support polygon of the humanoid in double support phase and dots indicate X-Y positions of centre of mass (COM) and circles represent position of end-effector, right hand in this case, during object manipulation by humanoid. The plot shows that the COM of humanoid stays inside the support polygon during task execution, which ensures the static stability of humanoid.

Table. 1 illustrates the optimization of cost achieved (*end-effector* path) with varying number of iterations.The mean path cost ± standard deviation are presented with varying number of iterations each carried out for 20 trials. As the number of iterations are increased, it can be seen that the average path cost is reduced, which is the expected result as RRT* is proven to be asymptotically optimal in [8]. In the experiment, final configuration is allowed to be within 0.03 units from required goal position. The time requirement for the path planning for given task is between 40 - 220 seconds for iterations ranging from 1500 - 4500. This time requirement increases with the increase in complexity of task i.e., when more obstacles are present or when task requires large movement in body. To counter this, a two tree version of RRT's similar to [18] can be used which is proven for its efficient exploration of configuration space.

Table 1: Experiment Results

Number of Iterations	Path cost(meters)
1500	1.71±0.33
2000	1.60±0.24
2500	1.54±0.24
3000	1.43±0.22
3500	1.34±0.15
4000	1.32±0.20



(a) (b) (c) (d)

Figure 5: (a)(b)(c) are simulation snapshots of humanoid placing a spherical object on table. (d) position of end-effector(circles) and COM(dots) during the task.

5. CONCLUSIONS

In this paper we presented an approach based on RRT* for optimal whole body motion planning of a humanoid with articulated spine. We have been able to optimize the end-effector path in terms of path-length in reaching the goal position. The loop-closure constraints were tackled by dividing the biped kinematic chain into two simple sub-chains and solving for loop-closure using forward kinematics on one chain and inverse kinematics on the other. The results show that these methods have been successful in manipulating the object without any constraints being violated.

Although the cost function considered in our approach is the Cartesian distance covered by end-effector during task, other important performance indices like total control effort or energy required can also be optimized using the same method. In future work, we will minimize the total control effort of humanoid using RRT* and compare our results with that of the Linear Quadratic Regulator control techniques. It has been observed after sequence of experiments, that a single tree version of RRT* is not sufficient for motion planning involving complex tasks. So, the future work will involve algorithm similar to connect version [18]. This involves two trees growing simultaneously, one from initial configuration and other from goal configuration, with balance in exploration and exploitation of configuration space.

6. REFERENCES

[1] B. Siciliano and J.J.E. Slotine. "A general framework for managing multiple tasks in highly redundant robotic systems". In ICAR 1991.

[2] O. Kanoun, F. Lamiraux, P.-B. Wieber, F. Kanehiro, E. Yoshida, and J.-P. Laumond, "Prioritizing linear equality and inequality systems:application to local motion planning for redundant robots" in ICRA 2009.

[3] Vande Weghe, M and Ferguson, Dave and Srinivasa, Siddhartha S, "Randomized path planning for redundant manipulators without inverse kinematics",7th IEEE-RAS *International Conference on Humanoid Robots*, 2007

[4] Bertram, Dominik and Kuffner, James and Dillmann, Ruediger and Asfour, Tamim, "An integrated approach to inverse kinematics and path planning for redundant manipulators",in ICRA 2006.

[5] S.M.LaValle andJ. J.Kuffner. "Randomized kinodynamic planning."in ICRA 1999.

[6] Jaillet, L., CortÃl's, J., SimÃl'on, T. "Transition-based RRT for path planning in continuous cost spaces." In IROS 2008.

[7] Bruce, J., Veloso, M. (2002). "Real-time randomized path planning for robot navigation." In IROS 2002.

[8] Karaman, S., Frazzoli, E. (2011). "Sampling-based algorithms for optimal motion planning". The International Journal of Robotics Research.

[9] Perez, Alejandro and Karaman, Sertac and Shkolnik, Alexander and Frazzoli, Emilio and Teller, Seth and Walter, Matthew R " Asymptotically-optimal path planning for manipulation using incremental sampling-based algorithms." in (IROS) 2011.

[10] Dalibard, S., Nakhaei, A., Lamiraux, F., Laumond, J. P. (2009, December). "Whole-body task planning for a humanoid robot: a way to integrate collision avoidance." In 9th IEEE-RAS *International Conference on Humanoid Robots(pp. 355-360)*. IEEE.

[11] Burget, F., Hornung, A., Bennewitz, M. "Whole-body motion planning for manipulation of articulated objects." In ICRA 2013.

[12] Vahrenkamp, N., Berenson, D., Asfour, T., Kuffner, J., Dillmann, R. "Humanoid motion planning for dual-arm manipulation and re-grasping tasks."In IROS 2009.

[13] L. Han and N. Amato. "A Kinematics-Based Probabilistic Roadmap Method for Closed Kinematic Chains." In *Workshop on the Algorithmic Foundations of Robotics*, 2000.

[14] Cortes, J.Simeon, T. ; Laumond, J.P. " A random loop generator for planning the motions of closed kinematic chains using PRM methodsAutom." in ICRA 1999.

[15] J. Nielsen and B. Roth. "Formulation and Solution for the Direct and Inverse Kinematics Problem for Mechanisms and Mechatronic Systems." In Proceedings of the NATO Advanced Study Institute on Computational Methods in Mechanisms, vol 1, pp. 233-252,1997

[16] Lapeyre, M., Rouanet, P., Oudeyer, P. Y. (2013, November). The poppy humanoid robot: Leg design for biped locomotion. In IROS 2013.

[17] Spong, M. W., Hutchinson, S., Vidyasagar, M. (2006). Robot modeling and control (Vol. 3). New York: Wiley.

[18] Kuffner, J. J., LaValle, S. M. RRT-connect: An efficient approach to single-query path planning.In ICRA 2000.