On Blocking Collisions between People, Objects and other Robots

Kwan Suk Kim, Luis Sentis

Abstract—Intentional or unintentional contacts are bound to occur increasingly more often due to the deployment of autonomous systems in human environments. In this paper, we devise methods to computationally predict imminent collisions between objects, robots and people, and use an upper-body humanoid robot to block them if they are likely to happen. We employ statistical methods for effective collision prediction followed by sensor-based trajectory generation and real-time control to attempt to stop the likely collisions using the most favorable part of the blocking robot. We thoroughly investigate collisions in various types of experimental setups involving objects, robots, and people. Overall, the main contribution of this paper is to devise sensor-based prediction, trajectory generation and control processes for highly articulated robots to prevent collisions against people, and conduct numerous experiments to validate this approach.

Index Terms—Collision prediction, robots blocking other robots, robot safety.

I. INTRODUCTION

Is it okay for robots to stop objects or other robots that could collide with people? Although ground robots and autonomous cars have operated for a while in human populated environments, it is unusual to see them intervening over collisions that might happen externally to their current path.

With today’s advancements on autonomous systems, we are prompted to study such a problem, that of measuring the probability of an accident about to happen followed by a decision process for a robot to stop the likely event when physically possible. A posit here is that a robot might produce benefits to humans from three levels of safety: i) collision avoidance when possible, ii) in the case that collision avoidance fails, then collision detection and fast contact reaction, iii) in the case that a collision between an external object or robot and a person is likely to happen, then block the collision if physically possible. In this paper we explore case iii). And we have first studied this case from an engineering perspective without entering into ethical, moral, medical, psychological or social questions. These other questions are of course very important, but we wanted first to understand what methods could be devised and employed to intervene over external events using human-centered robots. For this study we use an upper body humanoid robot consisting of an articulated torso and two articulated arms. For sensing we use a structure light sensor mounted outside of the blocking robot. The point cloud sensor is able to simultaneously “see” the objects in the scene, the humanoid upper body robot, and people nearby.

In order to make this study possible, the humanoid upper body robot needs two endow two capabilities: an estimation of the risk of a collision between nearby people and objects or robots, and an intervention strategy that is likely to block the object or robot if physically possible. We further give a twist to this study by considering that the blocking robot might be engaged into a task prior to considering intervening over a likely external collision. In that case, the robot must face the task of preventing the accident without stopping the task at hand. This question results in a study on constrained motion planning to engineer possible responses. Further details are broadly discussed next.

First, we use a point cloud sensor and prediction methods to estimate the risk that an object might collide with a person. Observing the current state of the surrounding objects to predict the future movements, the probability of collision of two objects or an object and a person is computed and the robot is controlled to prevent that collision from happening given a practical threshold risk value. Second, after it is decided that a collision prevention move should be attempted, it is necessary to compute intervention paths between any body part of the robot across its entire body and the estimated object path. For this purpose, we incorporate to our method a sampling-based motion planning algorithm and an analysis of the reachable volumes to determine the most favorable part of the robot’s body and its path to intervene over the object’s trajectory. We complement this computational processes with various
experiments involving the humanoid upper body mentioned before blocking moving objects or robots using the humanoid robot’s elbow, shoulder, torso, and end-effector in scenes where a human is likely to be struck by an object or by another robot. In light of this discussion, the main contribution of this paper is on devising sensor-based methods to effectively block collisions between external objects, robots and people using any part of a highly articulated robot, and while performing secondary tasks if physically possible.

II. BACKGROUND REVIEW

Many studies have been devoted to robot safety, such as [24] [28] [8], focusing on mitigation methods between colliding robots and humans. However, these studies do not consider blocking collisions between external objects or external robots and people. The likelihood that a robot will collide with its environment has been widely studied to reason about the degree of safety and to create control policies for collision mitigation [6] [29] [3] [5] [17] [4] [19]. Unlike our study the collisions anticipated in these studies are bounded to collisions between robots and their environments, without focusing on third-party collisions outside to a robot’s path.

Robots approaching moving objects have been well studied by the robot grasping community [18] [22] [11]. To grasp objects, a robot needs to estimate future object configurations. Some researchers predict future object configurations using a model of the object’s dynamics [22] [11]. But thanks to the recent innovations in machine learning, future object configurations can now be estimated without a prior model and used in a real-time process for grasping [14]. In contrast with grasping moving objects, our methods do not consider a particular appendage to interact with objects but instead perform a search over the entire articulated body of the robot.

As a twist on the research based on expected chores of human-centered robots, we assume that our blocking robot has a primary goal task prior to blocking moving objects. The control policy will attempt to find any other body part besides the task end-effector to block the moving object when physically possible. We will achieve this capability using sampling-based motion planning methods [10] [15]. To fulfill a task priority, we will further employ the constrained motion planning method [30] [2] [9] [11]. These methods have not been previously used to stop external objects or robots. Finally to perform these trajectories we will use whole-body operational space controllers on the intervening robot to enforce the task constraints via prioritized feedback linearization [25].

III. COLLISION PREDICTION

A. Object dynamics

To simplify the prediction process, we consider only the positions of the observed objects, reducing them to simplified circular/spherical rigid bodies with no rotation. Then, we determine the configuration of each object via its center position. We assume that the linear acceleration of objects is produced by an unknown source with zero-mean normal distribution. The state of the i-th object at the m-th time slice (we assume a temporal discretization) is defined as follows.

\[ p^i_m \triangleq \begin{pmatrix} p^i_{0,m} \\ \dot{p}^i_{0,m} \end{pmatrix} \]  

(1)

Also, each object is approximated with following dynamic equations including unknown disturbances \( \xi_p \).

\[ p^i_m = \begin{pmatrix} I & \Delta t I \\ 0 & I \end{pmatrix} p^i_{m-1} + \xi_p \]  

(2)

We assume that the position of the i-th object at time index \( m \), \( y^i_m \), has noise, \( \xi_s \), with a normal distribution as follows.

\[ y^i_m = \begin{pmatrix} I & 0 \end{pmatrix} p^i_m + \xi_s \]  

(3)

Then, the state of the i-th object and its covariance matrix can be estimated via a Kalman filter as follows.

\[ \hat{p}^i_m = \hat{p}^i_{0,m} + K^i_m (y^i_m - C_p \hat{p}^i_m) \]  

(4)

\[ \Phi^i_m = (I - K^i_m C_p) \Phi^i_m \]  

(5)

where all new variables are defined in Table I.

We define a new random variable to estimate the future state and its state transition equation, Eq. (2), the random variable \( X^i_k \) can be expressed using the following normal distribution.

\[ X^i_k \sim N \left( \hat{p}^i_m, \Phi^i_m \right) \]  

(6)

Given the initial state and its state transition equation, Eq. (2), the random variable \( X^i_k \) can be expressed using the following normal distribution.

\[ X^i_k = \begin{pmatrix} x^i_k \\ \dot{x}^i_k \end{pmatrix} \sim N \left( \begin{pmatrix} \bar{X}^i_{k-1} \\ \Sigma_{x,v,k} \end{pmatrix}, \begin{pmatrix} \Sigma_{x,k} & \Sigma_{x,v,k} \\ \Sigma_{x,v,k}^T & \Sigma_{v,k} \end{pmatrix} \right) \]  

(7)

where \( \bar{X}^i_{k-1} \) is the expected object state. In the state prediction, the time index \( k \) at the current time is set to zero. In Appendix B-C the random variables are approximated with normal distributions for computational efficiency.

We now define a conditional random variable, \( X_f^{ij} \triangleq (x_f^j, \dot{x}_f^j) \), representing the state of the i-th object under the condition that it has not collided yet with the j-th object. The probability distribution of the random variable is the complement of the collision probability between the i-th and j-th objects, so \( X_f \) is not a normal distribution. \( X_f \) also propagates in time similarly to \( X \) in Eq. (7) such that we can define another random variable, \( \hat{X}_f \) which is a one-step propagation process given \( X_f \). Even though \( X_f \) and \( \hat{X}_f \) are not normal distributions, the probabilistic properties of \( \hat{X}_f \) can be derived from those of \( X_f \) as shown in Appendix B-B.

\[ \begin{align*}
\mathbb{E} \left( \hat{X}_{f,k} \right) &= A_p \mathbb{E} \left( X_{f,k-1} \right) \\
\mathbb{V} \left( \hat{X}_{f,k} \right) &= A_p \mathbb{V} \left( X_{f,k-1} \right) A_p^T + \Sigma_w
\end{align*} \]  

(8)
B. Instantaneous collision probability

Based on the object dynamics of Eq. (2), we can predict the future locations of moving objects and also determine whether a collision external to a blocking robot will take place. By observing the current state of objects and their covariances, we can anticipate their future state, and estimate the possibility of their collision as stochastic processes as shown in Eq. (7).

For convenience, all the probability density functions and probabilities are denoted as \( P \) and \( p \), respectively, and all the predicates are denoted as \( q \). We define a probability function, \( P^i_o : \mathbb{R}^{N_o} \rightarrow \mathbb{R} \) being the probability that the center of the \( i \)-th object is located at a given point, \( p^i \), as follows.

\[
P^i_o \left( p^i \right) \triangleq P \left( x^i = p^i \right) \tag{9}
\]

where \( x^i \) is the center position of the \( i \)-th object. Also, a predicate prescribing whether the \( i \)-th and \( j \)-th objects collide with each other is labeled as \( q_{ij}(i,j) \).

Before computing the cumulative collision probability over time, we need to consider the probability of collision at a given time and the probability density function associated with object positions. The probability that the \( i \)-th object located at \( p^i \) collides with the \( j \)-th object can be derived from the probability density function of the \( i \)-th object, \( P^i_o \) and that of the \( j \)-th object, \( P^j_o \), and it can be expressed as \( P_{ij} \) as follows.

\[
P_{ij} \left( p^i, p^j \right) \triangleq P_{ij} \left( x^i = p^i, x^j = p^j \right) = P^i_o \left( p^i \right) P^j_o \left( f_{ij} \left( p^i, p^j \right) \right) \tag{10}
\]

\[
P_{ij} \left( p^i, p^j \right) = P^i_o \left( p^i \right) \int_{S} f_{ij} \left( p^i, p^j \right) P^j_o \left( dp^j \right) \tag{11}
\]

where \( f_{ij} \) is a function that checks collision events between the \( i \)-th and \( j \)-th objects, mapping the positions of the two objects to value 1 if the objects collide with each other or 0 otherwise. The probability density function, \( S \), corresponds to the probability that two objects collide with each other. The collision probability can be determined by the density functions at a given time instance, so we call it the **instantaneous collision probability**, \( P_{ij}^c \):

\[
P_{ij}^c \triangleq \int_{S} P_{ij} \left( p^i, p^j \right) \, dp^j \tag{12}
\]

We have derived the probability that two objects collide with each other. Therefore, we can also derive the probability that they do not collide with each other, and the conditional density function of the collision-free object positions. The collision density function of Eq. (10) is the probability that the \( i \)-th object located at \( p^i \) collides with the \( j \)-th object. The corresponding probabilistic density function describing that the \( i \)-th object at \( p^i \) is free from colliding with the \( j \)-th object is represented as \( P_{ij}^f \) as follows.

\[
P_{ij}^f \left( p^i, P^i_o, P^j_o \right) \triangleq P_{ij} \left( p^i \right) - P_{ij}^c \left( p^i, P^i_o, P^j_o \right) \tag{13}
\]

\[
P_{ij}^f \left( p^i, P^i_o, P^j_o \right) = P_{ij} \left( p^i \right) \int_{S} f_{ij} \left( p^i, p^j \right) P^j_o \left( dp^j \right) \tag{14}
\]

where \( P_{ij}^f \) is the collision-free probability density function at time \( k \) which is predicted from that at time \( k-1 \). From the definition of \( X_f \) and \( \hat{X}_f \) in Sec. III-A, \( P_{ij} \) and \( P_{ij}^f \) correspond to their probabilistic density functions.

\[
X_f \sim P_{ij}^f, \quad \hat{X}_f \sim P_{ij}^f
\]

If \( P_{ij}^f,k \) is a normal distribution, \( \hat{P}_{ij,k} \) can be estimated from Eq. (13). Though \( P_{ij}^f \) is not a normal distribution, we approximate it to be a normal distribution in Appendix B-C.

C. Cumulative collision probability

The collision probability given in Eq. (11) estimates only the instantaneous probability at a given time, so we cannot use that probability to determine how likely a collision happens at that time because a collision might already have happened and the objects involved in it could keep colliding. Thus, we need to recursively accumulate the collision probability and exploit the conditional probability that collisions do not happen before a time horizon. The predicate that describes this situation can be defined as follows.

\[
q_{ac,k}(i,j) \triangleq \bigvee_{n \leq k} \left\| x^i_n - x^j_n \right\| \leq r^i + r^j \tag{15}
\]

The corresponding accumulated collision probability is expressed and computed as follows.

\[
P_{ac,k}^{ij} = P \left( q_{ac,k}(i,j) \right) = P_{ac,k-1}^{ij} + \left( 1 - P_{ac,k-1}^{ij} \right) P \left( q_{ac,k}(i,j) \mid \neg q_{ac,k-1}(i,j) \right) \tag{16}
\]

The conditional probability on the last term of Eq. (16) is equivalent to the integral of the collision probability of Eq. (10), yielding the following.

\[
P_{ac,k}^{ij} = P_{ac,k-1}^{ij} + \left( 1 - P_{ac,k-1}^{ij} \right) \int_{S} P_{ij} \left( p^i, P_{ij}^f, P_{ij}^f \right) \, dp^i \tag{17}
\]
The cumulative collision probability, $p_{ac}$, increases over time and can be computed recursively starting at time zero. However, the derivation of the cumulative collision probability, $P_{ac}$, based on non-normal distributions is computationally expensive, prompting us to use an approximation. Thus, the probability is simplified for real-time computation purposes in Appendix B-A. Given the cumulative collision probability and a probability threshold, $\eta$, we can derive the minimum time index, $k_c$, at which the probability of a collision between any two objects from taking place exceeds the threshold as follows.

$$k_c = \arg\min_k$$

such that

$$\bigvee_{i,j \leq N_o} \left( (i \neq j) \land \left( p^{ij}_{ac,k} \geq \eta \right) \right)$$

(19)

$k_c$ embodies the likelihood that at least one object pair in the environment will collide with each other at time $k_c \cdot \Delta t$ with a probability $\eta$.

IV. BEHAVIOR PLANNING

To respond to the anticipated collisions, the robot agent uses four operating modes: idle, intervention, caution, and return. In the idle mode, the agent keeps predicting collisions and continuously planning a potential motion to block the most likely collision, but does not start an intervention maneuver in this mode. In the intervention mode, the agent uses the most favorable body part to intercept the likely colliding external objects with the hope to deviate one of them from the set trajectory. In the caution mode, the agent expects that anticipated collisions are unlikely but have enough risk. It then creates approaching behaviors to the threatening objects with its body parts without touching them. Finally, in the return mode, the agent assumes that there are no collision threats and returns to its normal operation. The modes and their transitions are described in the state diagram of Fig. 4.

A. Idle mode

The cumulative probability of a collision at every instant over a prediction horizon asymptotically increases as suggested by Eq. [19]. However, the time when the probability exceeds a given threshold depends on the initial object states and the statistics of the random variables. If a collision will happen in the distant future, the agent does not need to be concerned. Otherwise, the robot agent, according to premises, will intervene and attempt to stop the collision between the two external objects. Therefore, a probability threshold and a time threshold to differentiate the distant and the near futures need to be considered to reason about interventions. In Fig. 3 the probability threshold, $\eta$ and the time threshold, $T_{th}$ are characterized. If the collision probability of any two object at $T_{th}$ exceeds the probability threshold, $\eta$, then the agent intervenes by switching its mode from idle to intervention mode. Therefore, if any pair of objects are expected to collide with each other within the set time period, the mode of the robot agent changes as follows.

$$\left( \bigvee_{i,j \leq N_o, i \neq j} \left( p^{ij}_{ac,T_{th}/\Delta t} \geq \eta \right) \right)$$

$\rightarrow \text{(mode} \leftarrow \text{INTERVENTION)}$ (20)

The pair of objects for which the collision probability exceeds the threshold at $T_{th}$ forms a new set, $\mathcal{P}_c$, corresponding to the set of imminent collisions as follows.

$$\mathcal{P}_c = \left\{ (i,j) \left| p^{ij}_{ac,T_{th}/\Delta t} \geq \eta \right. \right\}$$

(21)

B. Intervention mode

In the intervention mode, the agent attempts to intercept the trajectory of the most imminent collision event. The most imminent collision is chosen from the set $\mathcal{P}_c$ defined in Eq. (21) by considering the positions and velocities of the external
objects. Each potential collision consists of two objects, and the time when these two objects will make contact or come
the closest to each other can be derived from their initial state.
Let the position and velocity of the i-th object be $p_i$ and $v_i$,
respectively, then the time when the distance between them is
minimum can be expressed as follows.

$$
\frac{d}{dt} \left( p_i + v_i \cdot t - (p_j + v_j \cdot t) \right) = 0 \tag{22}
$$

$$
(p_i - p_j) \cdot (v_i - v_j) < 0 \tag{23}
$$

The time at which the i-th and j-th object approach the closest
or collide with each other is defined as $t_{ij}^c$ and can be derived
analytically as follows.

$$
t_{ij}^c = \frac{(p_i - p_j) \cdot (p_i - p_j)}{(p_i - p_j) \cdot (v_i - v_j)} \tag{24}
$$

Then the pair of objects that will be involved in the most
imminent collision can be derived as follows.

$$
(i_c, j_c) = \arg\min_{(i, j)} t_{ij}^c, (i, j) \in \mathcal{P}_c \tag{25}
$$

Once the pair of objects involve in the most imminent collision
is identified, the positions and velocities of the objects are
used to determine the robot agent’s motion to intercept them.
From that pair, the time when they come the closest to each
other is derived via Eq. (24). Subsequently, a motion planner
described in the next section generates a path for the robot
agent to intercept the line segments of the external moving
objects along the colliding trajectory to be executed. Since
the most imminent collision is identified with our method and
the robot agent’s mode switches to the intervention mode, the
agent keeps updating the expected trajectories of the external
objects and determines whether to remain in the intervention
mode. The decision to stay in that mode considers the collision
probability and the inner product of the relative position and
velocity of the two objects. If the probability is still higher than
the threshold or the inner product is negative - meaning that the
distance between the objects decreases - then the agent remains
in the intervention mode. Otherwise, the mode switches to
cautious.

$$
\left( \bigwedge_{i, j \leq N_\text{obj}, i \neq j} \left( P^{ij}_\text{obj} < \eta \right) \wedge (d_{ij} \cdot v_{ij} \leq 0) \right) \rightarrow (\text{mode} \leftarrow \text{CAUTION}) \tag{26}
$$

Once the mode switches to intervention status and the most
imminent collision is determined, the external object pair
expected to be intercepted by the robot agent will not change
until the estimated collision probability becomes unlikely due
to the blocking effect of the interception. The agent will fully
complete the intervention for a particular collision even though
a more imminent collision is newly identified.

C. Caution mode

If the collision probability between the two external objects
is lower than the set threshold and the two objects are
determined to move away from each other, we assume that
a collision is unlikely. If this happens during the idle mode,
the agent will remain on that state. However, if it happens
during the intervention mode, the robot agent will change its
policy to caution mode because the intervention process in
which it is involved in will be less urgent.

Instead of "forgetting" the intervention task the agent
started, it keeps being "concerned" about it by using the caution
mode. In the caution mode, the agent neither intervenes
nor returns to fully focusing on the primary goal task. Instead
it stays ready for intervention and if all of a sudden the
collision probability increases, the robot agent goes back to
the intervention task.

D. Return mode

When the likelihood of a collision is low within the set
time horizon, $T_{cau}$, based on Eq. (26) the agent goes back to
its normal operation consisting on executing some primary
task other than blocking collisions. To return to a set default
posture for normal operation, the robot switches its logic mode
to return. The transition condition to trigger the return mode
is shown below.

$$
(mode = \text{CAUTION}) \wedge \left( \bigwedge_{i, j \leq N_\text{obj}, i \neq j} \left( P^{ij}_\text{obj} < \eta \right) \wedge (t > T_C) \rightarrow (\text{mode} \leftarrow \text{RETURN}) \right) \tag{27}
$$

V. MOTION PLANNING

As stated, the robot agent is set to intervene over likely
external collisions by using the most favorable body part
available over its entire body, and based on the behavior
triggers described in Sec. IV. Although there are many types
of robot “anatomies” and types of appendages, in this study
we consider an articulated upper body torso with an anthropo-
omorphic. The structure has $N_j$ joints and $N_l$ links and is
kinematically redundant for performing end-effector tasks in
SE(3) or $\mathbb{R}^3$. In our formulation, the robot agent will attempt
to use its most favorable body part to block collisions between
pairs of external objects, and it will do so by exploiting its
redundant kinematic abilities. To fulfill a primary task
supposedly being performed with the end-effector, any of the
robot agent’s body parts other than the end-effector will be
explored to block external collision unless there is no other
choice other than giving up the primary task and employing the

\begin{center}
\begin{figure}
\centering
\includegraphics[width=\textwidth]{state_transition_diagram.png}
\caption{State transition diagram. Once the collision probability exceeds a given threshold, the blocking robot triggers an intervention mode.}
\end{figure}
\end{center}
Fig. 5: Shown here is a robot’s configuration in which an $i$-th link makes a contact with an external object of radius $R_o$.

end-effector. To achieve such behavior, we leverage the method of reachable volumes where we simultaneously consider the reachable space for all body parts, and search in the robot’s configuration space for postures that fulfill the primary and the blocking tasks.

Subsequently, a we employ a motion planner to generate configuration space trajectories that achieve the desired goals. Since we are dealing with multiple objectives, the reachable volumes and the trajectories are derived using constrained planners. Even though the agent has redundant degrees of freedom to exploit the null space of the end-effector task, reaching the collision target may still turn to be physically impossible. In that case, the robot agent will be prompted to give up on the end-effector task and attempt to block the collision without task constraints. Because of this dichotomy, we will both compute constrained and unconstrained reachable motion plans.

Due to the need for real-time interventional speed, both trajectory planning and execution should be achieved within a set time horizon that increases the chances for success. To achieve this, we split the motion planning process into an off-line and an on-line procedures similar to the two-phase motion planning process discussed in [10]. During the off-line process, reachable volumes of all joints and linkages of the blocking robot are generated, and the configurations that correspond to the volumes are interconnected as tree structures according to the probabilistic roadmap (PRM) method - see [10]. During roadmap generation, dynamic properties associated with the configurations are also computed.

A. Computing reachable volumes

To use any of the blocking robot’s body parts, their position needs to be evaluated during the motion planning process. We approximate each part as two hemispheres connected with a cylinder as shown in Fig. 5.

To enable real-time interventional performance while exploring the aforementioned kinematic combinatorics, we generate reachable volumes for each body linkage a priori as described in [20]. In contrast, our reachable volumes are generated by sampling feasible configurations rather than analytically computing the Minkowski sum of the reachable volumes. This approach allows us to deal with more complex shapes and joint limits. We first model each link as a line segment and generate reachable volumes as shown in Fig. 5. Adjacent Cartesian coordinates of the link positions are clustered, and a 3D cell array structure is generated similarly to [27]. Each cell represents a set of adjacent points in 3D space and consists of the configurations for which the body link positions are located in the cell. We use an octree [21] to efficiently represent the reachable volumes. There are $N_l$ octrees for the constrained space and another $N_l$ octrees for the unconstrained case. The computation time for building octrees is $O(\log N)$, which is considered as moderately expensive. The algorithm for building octrees is shown in Algorithm 1.

The sampled configuration fed into the Octree algorithm is first generated by the PRM method [10]. As mentioned, constrained configurations are those which keep the end-effector at a desired task location while unconstrained configurations do not impose any requirement. To consider both constrained and unconstrained options, we generate two different roadmaps. The algorithm for roadmap generation is shown in Algorithm 2. In there, the variable $Link$ consists of four tuples, $\text{from}$, $\text{to}$, $r$, $q$ corresponding to the position of one end of the link, the position of the other end, the radius of the link, and the configuration to which the link belongs to, respectively. Algorithm 2 is similar to Algorithm 1, but in the former, the sampled configuration is used to generate the reachable volumes. Also, in the case of constrained motion planning, we
Algorithm 1. Building octree

function BuildOctree( Link, Octree )
    p₁ ← Link.from
    p₂ ← Link.to
    r ← Link.radius
    if Octree.depth = MAX_DEPTH then
        Octree.links ← Link
        return
    end if
    if Octree.depth = ∅ then
        CreateChildren(Octree)
    end if
    for child in Octree.children do
        d ← getDistToLineSeg(child.center, p₁, p₂)
        if d ≤ child.width + r then
            BuildOctree( Link, child )
        end if
    end for
    return
end function

Algorithm 2. Learning phase

function Learn
    N ← ∅, E ← ∅, O ← ∅
    while 1 do
        q₁new ← RandomConf()
        q₁near ← NearestNeighbor(q₁new, N)
        dq ← Project(q₁near, q₁new - q₁nearest)
        Δq ← dq/|dq| × STEP
        q₁prev ← q₁near
        q₁ ← q₁prev + Δq
        if IsFeasible(q₁) do
            N ← N ∪ q₁
            E ← E ∪ (q₁, q₁prev)
        end if
        for l₁ in Links(q₁) do
            BuildOctree(l₁, O₁)
        end for
        q₁prev ← q₁
        q₁ ← q₁ + Δq
    end while
end function

B. Searching for intervening body parts

During the collision intervention process described in Sec. IV, the blocking robot needs to determine which body part can intercept the predicted external collision. Because the reachable volumes of all the linkages of the blocking robot are computed in advance, the feasible configurations for intervention can be determined by searching the reachable volumes. Given an object position, p₀, its velocity, v₀, and its radius, r₀, we can define the set of configurations, C₁, at which the j-th link intersects the trajectory as follows.

\[ C_{int,j} = \{ l_k.q ∈ O_j.links, \quad Distance(l_k.from, l_k.to, p_0, v_0 \cdot t_d) ≤ r_0 + r \} \]  

where the function, Distance returns the minimum distance between two line segments. The four arguments for that function are the pairs defining the line segments. The configurations of the previous set have at least one link that intersects the object trajectories. If the set has at least one element, the robot can intervene to stop the collision using the corresponding body link. Our robot consists of N₀ body linkages, and we consider constrained and unconstrained motion plans. Therefore, given a collision trajectory, 2N₀ sets of intersecting configurations need to be considered.

The decision policy for the intervening body linkage is shown in Fig. 7(a). Whenever the state of the external objects changes, the blocking robot determines new intervening body parts. If the current robot configuration intersects the trajectory of the object (Fig. 7(a)), then it should remains at that configuration. Otherwise, the decision process checks whether the blocking robot is operating in the intervention mode (Fig. 7(b)) such that it can decide whether to reuse the previous motion plan. If the robot is already in intervention mode, the final destination given by the previous motion plan is re-evaluated to see whether it still intersects with the collision trajectory (Fig. 7(c)). If it does, the planner reuses the previous motion plan. In the case that the final destination does not intersect the collision trajectory, we replan the intervention trajectory. To replan the trajectory seamlessly, we attempt to reuse the same intervening body linkage of the robot computed from the previous motion planning process (Fig. 7(d)). If the reachable volume of that linkage intersects the collision trajectory, we build a new motion plan to reach one of the configurations in the volume. Otherwise, we replan from scratch the entire task.

If the current state violates the end-effector task (Fig. 7(c)), we search within the unconstrained reachable volumes. Otherwise, we search within the constrained reachable volumes. If the search fails, the blocking task fails because the robot cannot intercept the collision trajectory.

C. Constrained motion planning

In this subsection, we consider the constrained manifold imposed by the primary end-effector goal task using whole-body operational space control (WBOSC). Thanks to the task hierarchy capability of WBOSC, we can generate a motion

exploit the null-space projection of the primary goal task to fulfill the primary end-effector task. The detailed explanation of the constrained planner is described in Sec. V.C. The constrained and unconstrained reachable volumes of our robot's left upper arm are shown in Fig. 6(c) and (d). Whenever a new sampled configuration is considered, the sample is registered as a new node in a search tree. In addition, the Cartesian coordinates of each link of the sampled configuration is also registered in a corresponding 3-dimensional cell of the reachable volume. The roadmap also provides paths to connect all configurations, so the motion plan from the current state to the desired interventional state can be effectively derived from the roadmap.
plan compliant with higher priority tasks. The task hierarchy of WBOSC is provided by null space projections associated with higher priority tasks. An introduction to WBOSC and detailed descriptions and derivations are provided in Appendix A.

The rank of $\Lambda_2^*$ in that appendix, which is equivalent to that of $\Lambda_2^+$, is the same as the dimension of the task space coordinates (e.g. the end-effector position). Thus, by the definition of $\Lambda_2^*$, the rank of $J_2'N_1^T\Phi^*N_1^TJ_2'T$ emerges from the following model.

$$\begin{align*}
\Lambda_2^* &= J_2'N_1^T\Phi^*N_1^TJ_2'T \\
&= J_2\Sigma\Sigma^TJ_2'T
\end{align*}$$

where $N_1$ is the null space projection of the higher priority task into the lower priority task, and is defined as follows.

$$N_1 = I - A^{-1}N_cJ_1'T (J_1A^{-1}N_cJ_1'T) + J_1N_c$$

Because $A$ is a positive definite matrix, we only check the rank of $J_2N_1$ to identify the dimension of the lower priority task. The rank of $N_1$ is determined from Eq. (30). Taking the singular value decomposition of $A^{-1/2}N_cJ_1'T$, Eq. (30) can be expressed as follows.

$$\begin{align*}
N_1 &= I - A^{-1/2}U\Sigma\Sigma^T (\Sigma\Sigma^T)^+ \Sigma\Sigma^T A^{-1/2} \\
&= A^{-1/2}(I - U\Sigma\Sigma^T)A^{-1/2} \\
&= A^{-1/2}(I - U\Sigma\Sigma^T)^+ A^{-1/2}
\end{align*}$$

If the rank of the diagonal matrix, $\Sigma \in \mathbb{R}^{r \times n}$ is $r < n$, then $\Sigma\Sigma^T + T\Sigma = \begin{pmatrix} I_r \times r & 0 \\ 0 & 0 \end{pmatrix}$ and Eq. (31) becomes the following.

$$N_1 = A^{-1/2} \left(I - U \begin{pmatrix} I_r \times r & 0 \\ 0 & 0 \end{pmatrix} U^T \right) A^{1/2}$$

Thus, the rank of $N_1$ is determined by the robot’s number of generalized coordinates minus the rank of $J_1N_c$. The constrained manifold can be derived by substituting Eq. (29) into Eq. (41) as follows.

$$\tilde{x}_2 = \Lambda_2^+\Lambda_2\tilde{x}_{2,des} + \tau_1''$$

Because $\Lambda_2^*$ is a symmetric matrix, we can take its singular value decomposition and express it as $U\Sigma\Sigma^T$. Then, $\tilde{x}_2$ can be derived as follows.

$$\begin{align*}
\tilde{x}_2 &= U\Sigma^+U^T\Sigma\Sigma^T \tilde{x}_{2,des} + \tau_1'' \\
&= U \begin{pmatrix} I_r \times r & 0 \\ 0 & 0 \end{pmatrix} U^T \tilde{x}_{2,des} + \tau_1''
\end{align*}$$

where $r_2$ is the rank of $\Lambda_2^*$. In this equation, $U$ consists of singular vectors, $\begin{pmatrix} u_1 & u_2 & \cdots & u_n \end{pmatrix}$ and $U^T\tilde{x}_{2,des}$ can be defined as an arbitrary vector, $z = \begin{pmatrix} z_1 & z_2 & \cdots & z_n \end{pmatrix}^T$. Then, $\tilde{x}_2$ can be expressed as the weighted sum of the independent basis as follows.

$$\tilde{x}_2 = z_1u_1 + z_2u_2 + \cdots + z_r u_r + \tau_1''$$

Therefore, the singular vectors of the task space mass matrix, $\Lambda_2^*$ are the basis of the constraint manifold considering the null space of higher priority tasks.

From the previous section, the basis of the lower priority task has been derived. With this basis, we can generate the
Fig. 8: The experimental setup consists of an external Kinect camera and the Dreamer robot (a) which consists of an upper-body humanoid robot. Dreamer consists of two 7-DoF arms a 3-DoF torso, the latter including two forward bending coupled joints and a vertical rotation. Each robot arm consists of 7 revolute joints. We approximate the forward bending torso with two capsules. The simplified stick figure for the robot (only considering the torso and left arm) is shown in (b). The transformation between the robot coordinates and the Kinect coordinates is calculated beforehand, so all the RGB points from Kinect can be represented in the robot’s coordinates as shown in (b).

Project function shown in Algorithm 1. The arguments of that function are the current robot configuration and the displacement of the joints. To make the displacement compliant with the end-effector task constraint, it needs to be projected into the null space of the end-effector based on Eq. (33). \( \Lambda_2^\tau \) is the dynamic term as a function of \( q \), such that the projected displacement shown in Algorithm 1 takes the form.

\[
\hat{x}_2 = \Lambda_2^\tau(q) \Lambda_2^\tau(q) (\hat{x}_{\text{disp}} - \tau') + \tau' \tag{36}
\]

VI. EXPERIMENTS

A. Experimental setup

To validate the proposed methods, we conduct collision intervention experiments using an upper-body humanoid robot, Dreamer, shown in Fig. 8. Detailed information about the robot can be found in [23].

To sense its environment, the robot uses a Kinect sensor to detect external objects. Objects in the robot’s environment are identified via a color profile and markers. The color profile of a tennis ball is considered in advance, and object markers are set to be tracked using the Aruco computational library [23]. Therefore, whenever the Kinect sensor receives a new RGB image frame with a corresponding depth image, the blocking robot can track the green tennis ball and the various registered markers such as the one shown in Fig. 10 while the locations and the velocities of the objects are estimated via Kalman filters as shown in Eq. 2. The parameters for the Kalman filter are shown in Table 1. Given those states, the collision probability between two objects can be estimated as shown in Sec. III. The whole experimental setup with Dreamer and the Kinect sensor is shown in Fig. 8.

The blocking robot, Dreamer, is set to perform a primary goal task in which the end-effector moves and remains at a stationary goal position. Therefore collision intervention tasks to stop external objects are attempted to be performed under the null space of the primary task when possible. The end-effector’s stationary goal is shown as a red ball in Fig. 2(a). Before a collision intervention task takes place, the robot generates the constrained and unconstrained probabilistic roadmaps using 10,000 samples, and the reachable volumes of the shoulder and arm linkages corresponding to the roadmaps are also generated, as shown in Fig. 2(a) and (b). The volume of the operating environment around the blocking robot is \( 2m \times 2m \times 2m = 8m^3 \), and we segment it via the Octree algorithm with 6 levels of depth. Therefore, we consider 8 reachable volumes corresponding to the constrained/unconstrained roadmaps and the four robot linkages. The dimension of each cube in the Octree is \( 3cm \times 3cm \times 3cm \). The generated reachable volumes of the constrained roadmap for the 4 linkages are shown in Fig. 9. If the object trajectory overlaps any of the constrained volumes, the blocking robot is able to simultaneously stop the external object’s movement while keeping the end-effector on its goal position. The search for this process is conducted based on the decision policy shown in Fig. 7.

B. Constrained collision intervention

In this experiment, the object with the marker represents a human subject working nearby the blocking robot, and the ball represents a moving object harmful to the subject. The objective of the blocking robot is to 1) observe the human and the ball, 2) determine whether these objects will collide with each other, and 3) stop potential collisions between these two objects by using one of its body parts. This scene is illustrated in Fig. 10. Object 1 is the ball, which approaches object 2, the

| Name | Description | Definition/Value | Metric |
|------|-------------|-----------------|--------|
| \( \xi_i^p \) | Random variable of state estimation noise | \( \begin{pmatrix} 0, \Sigma_d & 0 \\ 0 & \Sigma_u \end{pmatrix} \) | |
| \( \xi_s^i \) | Random variable of sensor noise | \( (0, \Sigma_s) \) | |
| \( \hat{P}_m^i \) | Predicted state | \( A_m \hat{P}_m^ {i-1} \) | |
| \( \Phi_m^i \) | Predicted covariance | \( A_m \Phi_{m}^{i-1} A_m^T + \Sigma_w \) | |
| \( S_m^i \) | Innovation covariance | \( C_m \Phi_{m}^{i-1} C_m^T + \Sigma_w \) | |
| \( K_m^i \) | Kalman gain | \( \Phi_{m}^{i-1} C_m^T \Sigma_w^{-1} \) | |
| \( \Delta t \) | Sampling time | 33 msec | |
| \( \Sigma_d \) | Covariance of velocity disturbance | diag(0.01,0.01,0.01) | \( (m/s)^2 \) |
| \( \Sigma_a \) | Covariance of acceleration | diag(1.5, 1.5, 1.5) | \( (m/s^2)^2 \) |
| \( \Sigma_s \) | Position sensor noise | diag(0.01,0.01,0.01) | \( m^2 \) |
The reachable volumes of the linkages corresponding to the constrained roadmap are generated for the occupancies of the shoulder, the upper arm, the lower arm, and the end-effector.

The blocking robot intercepts object 1 if it is determined that object 1 is set to collide with object 2. We assume that object 2 is a placeholder for a human operator who needs to be protected against object 1, which is a ball. The blocking robot tracks the marker and the green ball using Aruco and the OpenCV library. During the tracking process, the positions and velocities are estimated via Kalman filter. The Kinect sensor updates the input video sequence every 33msec and so does the Kalman filter.

Fig. 9: Concept image of the experiment. The blocking robot intercepts object 1 if it is determined that object 1 is set to collide with object 2. We assume that object 2 is a placeholder for a human operator who needs to be protected against object 1, which is a ball. The blocking robot tracks the marker and the green ball using Aruco and the OpenCV library. During the tracking process, the positions and velocities are estimated via Kalman filter. The Kinect sensor updates the input video sequence every 33msec and so does the Kalman filter.

Fig. 10: Concept image of the experiment. The blocking robot intercepts object 1 if it is determined that object 1 is set to collide with object 2. We assume that object 2 is a placeholder for a human operator who needs to be protected against object 1, which is a ball. The blocking robot tracks the marker and the green ball using Aruco and the OpenCV library. During the tracking process, the positions and velocities are estimated via Kalman filter. The Kinect sensor updates the input video sequence every 33msec and so does the Kalman filter.

The movement of the three objects including the blocking robot is estimated, and the probability that the external objects collide with each other is predicted based on Eq. (16). To determine whether there will be a collision or not, the threshold values defined in Fig. 5 needs to be set. In this experiment, we use $\eta = 0.5$ and $t_c = 4$, which means there will be a collision if the cumulative collision probability is 0.5 or above it and within a 5 seconds prediction horizon. If the collision probability exceeds the threshold, the robot searches its reachable volumes for postures intercepting the trajectory of the external objects as shown in Fig. 5. According to the search policy shown in Fig. 7, the blocking robot searches the constrained reachable volumes first. If it fails to find a feasible solution, it searches the unconstrained reachable volumes. If the robot finds an intercepting final pose, it will generate a motion path from the current pose to the intercepting pose given the PRM plan.

Three separate collision intervention processes are shown in Figs. 12 and 13. In all of these cases the first set of images depict the robot determining that objects 1 and 2 will collide against each other, and switching its state to the intervention mode. In the intervention case shown in Fig. 12 (a), the blocking robot intervenes using its upper arm, while in Fig. 12 (d) it determines that the best body part is using its lower arm. In addition, Fig. 13 shows an intervention process using the robot’s shoulder. Reachable volume data analysis is further shown in Fig. 11 for the first two experiments.

C. Unconstrained collision intervention

If the predicted external object’s trajectory does not overlap with any of the constrained reachable volumes of the blocking robot, the robot cannot intervene while satisfying the primary end-effector goal task. In the experiment shown in Fig. 14, the blocking robot is allowed to violate the goal task. Fig. 15 shows the constrained and unconstrained reachable volumes corresponding to that experiment, which justify the decision to violate the primary goal task. In the first image sequence of Fig. 14, object 1 approaches object 2, and when collision probability exceeds the designed threshold, the robot decides to intervene.
Fig. 12: **Collision intervention using the robot's upper and lower arm.** The robot generates a motion plan which makes its upper arm stop object 1 from colliding with object 2 (a). Subfigure (b) shows the posture and ball trajectories. Subfigure (d) shows the predicted collision probability at each time frame. At time 0.5, the probability exceeds the threshold value within 5 seconds, and therefore the blocking robot chooses to stop the likely collision. The blue arrow corresponds to the predicted object motion. The robot’s lower arm blocks object 1 from colliding with object 2 (d), and its trajectory and collision probability are shown in (e) and (f), respectively.
Fig. 13: Collision intervention using the robot’s shoulder. The reachable volume of the robot’s shoulder overlaps with the estimated ball trajectory and as a result the collision intervention process is chosen to occur using the shoulder.

Fig. 14: Unconstrained collision intervention using the robot’s lower arm. Here, there is no constrained reachable volume that can be used for collision intervention. As a result, the robot intervenes using its unconstrained lower arm.

D. Concept experiment blocking and external robot

Fig. 16 shows experiments involving interventions of the upper-body humanoid robot Dreamer to stop a ground mobile robot, Trikey, from colliding with a person. While the mobile robot approaches the standing person, the blocking robot estimates the probability of an external collision. If it exceeds a set threshold, the blocking robot intercepts the collision by stopping the mobile robot. In addition, the mobile robot incorporates a control algorithm that changes direction once it senses a contact as described in [13]. As a result, once touched, the mobile robot will move away on the opposite direction thus preventing injuring the human. This study assumes that every now and then collision avoidance on mobile robots will fail to work and other means of safety could potentially be beneficial like the one considered here.

VII. Conclusion

In this paper, we investigate methods for safety by blocking potential collisions between external objects or robots and people. We estimate collision probabilities and compared against set thresholds. Secondly, we generate motion plans to block likely collisions using an upper body humanoid robot. For intervention, we consider any body part of the entire robot as candidates for blocking the collision. We also assume that the robot is already engaged in a primary goal task using its end-effector, and we utilize the redundant degrees of freedom to stop external collisions when possible. We computationally build constrained reachable volumes for all robot’s body parts and interconnect corresponding postures via PRM planning. We then check whether the reachable volumes overlap with the collision trajectories. If any of the reachable volumes does overlap, a motion plan is generated to accomplish the intervention. Otherwise, the blocking robot searches for overlapping postures within the unconstrained reachable volumes. These methods are implemented in a set of robotic systems involving collisions between a ball, a mobile robot, and a human subject. As future extensions, fast visual sensors and time-optimal controllers could be adopted to reduce the response times. Also, our task planner could be improved by considering not only the collision probability but also the severity of collisions, and the importance of each object in the environment.

Appendix A

Whole-body operational space control

A robot’s dynamic equation incorporating contacts constraints has the following expression [12].

\[ A\ddot{q} + N_c^T bg = (UN_c)^T \tau \]

where \( A, g, N_c, bg, U, \) and \( \tau \) are the mass matrix, the generalized robot coordinates, the constraint null space projection associated with contacts, Coriolis-centrifugal-gravity terms, the selection matrix of actuated joints, and torque/force input to the robot, respectively. According to the WBOSC...
framework, the torque input for the highest priority task is computed as follows.

\[
A\ddot{q} + N^T bg = (UN_c)^T (J_1^* \Lambda_1^* \ddot{x}_{1,des} + N_1^* \tau_2) \\
+ (UN_c)^T (UN_c^T bg)
\] (38)

where \(J_1^*\), \(\Lambda_1^*\), \(N_1^*\), \(\tau_2\) are the Jacobian matrix of the highest priority task, the highest priority task’s mass matrix, the null space of the task, the dynamically consistent pseudo-inverse of \(\cdot\), and the torque input of lower priority tasks, respectively. If the robot is fully controllable under contact constraints, which means \(\bar{U}N_cU_c = N_c\), then \(bg\) can be canceled out as follows.

\[
A\ddot{q} = (UN_c)^T (J_1^* \Lambda_1^* \ddot{x}_{1,des} + N_1^* \tau_2) 
\] (39)

Lower priority tasks should be compliant with higher priority tasks, so the lower priority tasks are constrained. In Eq. (39), the lower priority task input \(\tau_2\) is projected into the null space of the higher priority task by left-multiplying by \(N_1^*\). \(N_1^*\) is defined as follows.

\[
N_1^* = I - J_1^* J_1^{**}
\] (40)

By left-multiplying \(J_2 A^{-1}\) by Eq. (39), the constraint dynamic equation becomes

\[
\ddot{x}_2 + J_2 \ddot{q} = J_2 A^{-1} (UN_c)^T (\tau_1 + N_1^* J_2^* T \Lambda_2^* \ddot{x}_{2,des}) \\
= \tau_1' + J_2^* D_1^* T \Lambda_2^* \ddot{x}_{2,des} \\
= \Lambda_2^* \ddot{x}_{2,des} + \tau_1'
\] (41)

If the lower priority task’s mass matrix, \(\Lambda_2^*\) is full rank, the task is fully controllable and \(\ddot{x}_2\) equals the desired task acceleration, \(\ddot{x}_{2,des}\).

**APPENDIX B**

**COLLISION PROBABILITY BETWEEN TWO OBJECTS**

**A. Probability of collision**

For easy understanding, a one-dimensional collision between two objects labeled as \(i\) and \(j\) is considered first. Let’s assume that there are two one-dimensional objects consisting of line segments with length \(l_i\) and \(l_j\), and their center positions are distributed with probability densities, \(P^i_o\) and \(P^j_o\). Then, the probability of a collision between them is described as follows.

\[
P_{ij}^o = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f_{ij}(x_i, x_j) P^i_o(x_i) P^j_o(x_j) dx_j dx_i
\]

\[
= \int_{-\infty}^{\infty} P^i_o(x_i) \left( \int_{x_i-w}^{x_i+w} P^j_o(x_j) dx_j \right) dx_i
\]

\[
= \int_{-\infty}^{\infty} P^i_o(x_i) \left( \int_{-w}^{w} P^j_o(x_i + x) dx \right) dx_i
\]

where \(x \triangleq x_j - x_i\) is the relative position of \(x_j\) with respect to \(x_i\), and \(f_{ij}\) is the predicate function which returns 1 if the two objects positions \(x_i\) and \(x_j\) end up colliding as follows.

\[
f_{ij}(x_i, x_j) = \begin{cases} 1 & -w \leq x_i - x_j \leq w \\ 0 & \text{otherwise} \end{cases}
\] (43)

where \(w \triangleq l_i + l_j\).

Using the Fubini theorem, the two integrals commute.

\[
= \int_{-w}^{w} \left( \int_{-\infty}^{\infty} P^i_o(x_i) P^j_o(x_i + x) dx_i \right) dx
\]

\[
= \int_{-w}^{w} \left( \int_{-\infty}^{\infty} P^i_o(x_i) P^j_o(-(-x - x_i)) dx_i \right) dx
\]

Defining a new function, \(P^j_o'\) such that \(P^j_o'(x) \triangleq P^j_o(-x)\), the probability becomes the integral of the convolution as follows.

\[
= \int_{-w}^{w} \left( P^j_o'(-x) * P^j_o'(-x) \right) dx
\] (45)
In the case of the one-dimensional scenario, the shape of objects are line segments, which are symmetric, so the integration of their Minkowski sum is also symmetric.

\[
\int_{-w}^{+w} \left( P_{o1}(x) \ast P_{o2}(x) \right) dx
\]  

The convolution of two normal distributions are also a normal distribution, and its mean and covariance are derived from those of the two normal distributions. So, provided that \( P_{o1} \) and \( P_{o2} \) are assumed to be normal distributions with the following properties.

\[
P_{o1}(x_i) = \mathcal{N}(\mu_i, \sigma_i^2) \\
P_{o2}(x_j) = \mathcal{N}(\mu_j, \sigma_j^2)
\]  

(47)

the convolution of the two normal distributions becomes another normal distribution, \( P_{\text{conv}} \) as follows.

\[
P_{\text{conv}}(x) = P_{o1}(x) \ast P_{o2}(x) \\
= \mathcal{N}(\mu_{ij}, \sigma_{ij}^2)
\]  

(48)

Therefore, the probability of collision between the two objects in one-dimensional space is the cumulative density function of the new normal distribution, and the instantaneous collision probability between the two objects, \( p_{ij} \), becomes a cumulative density function of a normal distribution as follows.

\[
p_{ij} = \int_{-\infty}^{\infty} P_{\text{conv}}(x) dx
\]  

(49)

Depending on the distance of the two normal distributions and the size of the objects, there are two ways to express the collision probability using the error function, \( \text{erf} \).

\[
p_{ij} = \begin{cases} \frac{1}{2} \left( \text{erf} \left( \frac{x^+}{\sqrt{\sigma_i^2 + \sigma_j^2}} \right) - \text{erf} \left( \frac{x^-}{\sqrt{\sigma_i^2 + \sigma_j^2}} \right) \right) & |\mu_i - \mu_j| > \infty \\ \frac{1}{2} \left( \text{erf} \left( \frac{x^+}{\sqrt{\sigma_i^2 + \sigma_j^2}} \right) + \text{erf} \left( \frac{x^-}{\sqrt{\sigma_i^2 + \sigma_j^2}} \right) \right) & |\mu_i - \mu_j| \leq \infty \end{cases}
\]  

(50)

where \( x^+ \triangleq \frac{|\mu_i - \mu_j| + \infty}{\sqrt{\sigma_i^2 + \sigma_j^2}} \) and \( x^- \triangleq \frac{|\mu_i - \mu_j| - \infty}{\sqrt{\sigma_i^2 + \sigma_j^2}} \).

To extend collisions in one-dimensional space to 3D space, the overlapping predicate function of Eq. (43) also needs to be extended. If it is assumed that the objects are rigid bodies, then the predicate depends only on their relative position and we can define another predicate function associated with the relative position, \( F_{ij} \), as follows.

\[
F_{ij}(x) \triangleq f(0, x)
\]  

(51)

Because the predicate only depends on the relative position of the two objects, it has the following property.

\[
\forall x, F_{ij}(x) = f(x_i, x_i + x)
\]  

(52)

Also, we can define a Minkowski sum of the two objects as \( B \), and the integral of an arbitrary function over the region has the following property.

\[
\int_B g(x) dx = \int_{-\infty}^{\infty} F_{ij}(x) g(x) dx
\]  

(53)

Then, the one-dimensional collision probability in Eq. (42) is extended to the 3D case as follows.

\[
\int_{-\infty}^{\infty} P_{ij}(x_i) \left( \int_B P_{o1}(x_i + x) dx \right) dx_i = \int_B \left( P_{o1}(-x) \ast P_{o2}(-x) \right) dx
\]  

(54)

where \( P_{o1} \) and \( P_{o2} \) are joint probability density functions in 3D with the following properties.

\[
P_{ij}(x_i) = \mathcal{N}(\mu_i, \Sigma_i) \\
P_{o1}(x_j) = \mathcal{N}(\mu_j, \Sigma_j)
\]  

(55)

The convolution of the normal distribution in 3D space is also a joint normal distribution. So, the collision probability in 3D is also a cumulative probability function of a normal distribution as follows.

\[
p_{ij} = \int_B P_{\text{conv}}(x) dx
\]  

(56)

where \( P_{\text{conv}} \) is the convolution of the two normal distributions which is another normal distribution with the following property.

\[
P_{\text{conv}} \sim \mathcal{N}(\mu_i - \mu_j, \Sigma_i + \Sigma_j)
\]  

(57)

In this paper, we assume that all objects are spheres and non-rotating, so the Minkowski sums are always ellipsoid. Therefore, the collision probability is derived from the integration of the normal distribution around an ellipsoid. A computationally efficient integration algorithm is considered from [26] to compute the collision probability.

**B. Conditional probability density function of collision**

From Eq. (42), the probability that the i-th object located at \( p^i \) collides with the j-th object in 3D can be described as follows.

\[
p_{ij} = \int_{-\infty}^{\infty} f_{ij}(p^i, p^j) P_{ij}^o(p^o) dp^o
\]  

(58)

where \( f_{ij} \) and \( F_{ij} \) are the predicate functions defined in Eqs. (43) and (51). Using the probability distribution, the corresponding random variable for the first object, \( X_{c,1} \), can be defined. We can assume that the probability of the second object \( P_2 \) is a normal distribution, but \( F_{ij} \) is obviously a function that indicates the Minkowski sum of the two objects. However, based on the central limit theorem that repeated convolutions of two probability density functions converge to a normal distribution, we can assume that the convolution can be approximated to a normal distribution.

To derived the properties of the approximated normal distribution, we use the real mean and the variance of the convoluted function as follows.

\[
E(X_{c,1}) = \int_{-\infty}^{\infty} p^i P_{ij}^o(p^j) dp^i
\]

(59)
The denominator becomes the volume of the Minkowski sum of the two objects, $V_B$, because the integral of the convolution in Eq. (58) becomes the product of the integrations of both functions.

By defining a new variable $\eta \triangleq p^i - p^j$, we can further simplify as follows.

$$\frac{1}{V_B} \int_{-\infty}^{\infty} P_2(p^i) \int_{-\infty}^{\infty} (p^i - \eta) F_{ij}(\eta) d\eta d\eta = \frac{1}{V_B} \int_{-\infty}^{\infty} P_2(p^i) \int_{-\infty}^{\infty} F_{ij}(\eta) d\eta - \int_{-\infty}^{\infty} P_2(p^i) dp^i \int_{-\infty}^{\infty} \eta F_{ij}(\eta) d\eta = \mu_2 - \frac{1}{V_B} \int_{-\infty}^{\infty} \eta F_{ij}(\eta) d\eta$$

We can define that the center of the Minkowski sum $F_{ij}$ located at the origin as mentioned above.

Therefore, the covariance of the conditional variable, $X_{c,1}$, can be simplified as follows.

$$\text{cov}(X_{c,1}) = \Sigma_2 + \frac{1}{V_B} \int_{-\infty}^{\infty} \eta^2 F_{ij}(\eta) d\eta$$

The second term becomes zero because the center of $F_{ij}$ is located at the origin as mentioned above.

Therefore, the conditional covariance of $X_{c,1}$ can be simplified as follows.

$$\text{cov}(X_{c,1}) = \Sigma_2 + \frac{1}{V_B} \int_{-\infty}^{\infty} \eta^2 F_{ij}(\eta) d\eta$$

Then, we can define a new random variable, $\tilde{X}_c$, with a normal distribution and its statistic properties are the same to those of $X_c$ as follows.

$$\tilde{X}_{c,1} \sim \mathcal{N}(\mu_2, \Sigma_2 + C_B/V_B)$$

C. Implementation considerations

To predict an imminent collision with time index, $k_c$, we would ideally need to compute the instantaneous collision probability of objects using non-Gaussian distributions according to Eq. (11). This computation would take a large amount of time to finally predict the collision-free object distributions of Eq. (12). Sampling-based probability derivations have been proposed for computational efficiency [16], however these computations remain expensive for real-time usage. Therefore, we compromise on the accuracy of the estimation using other simplifications.

First, we simplify the instantaneous collision probability of Eq. (11), which consists of an integration over the considered environment and the shapes of the objects on it. By assuming the distribution of the objects to be normal, the computation becomes efficient. The integration operation then becomes a single normal distribution, which allows the use of pervasive efficient numerical solutions for the integration of multivariate normal distributions. The detailed approximation and computation of $p_{ic}$ are explained in Appendix B-A. If the object distributions are assumed to be normal, the collision probability of the two objects can be simplified as follows.

$$p_{ij}^{(e)} = \frac{1}{V_B} \int_{B_{\text{conv}}} p(x) dx$$

where $B_{\text{conv}}(\cdot) d\eta$ denotes the integration along the Minkowski sum of the $i$-th object and $j$-th object, and $p_{\text{conv}}$ is a normal distribution with the following probability.

$$p_{\text{conv}} \sim \mathcal{N}(\mu_i - \mu_j, \Sigma_i + \Sigma_j)$$

The approximate integration of multivariate normal distributions around elliptic shapes is studied in [26]. Therefore, the instantaneous collision probability can be computed effectively, such that the conditional probability on the last term of Eq. (13) can be simplified.

Second, we revisit the conditional probability density function $P_{of}$ of Eq. (12), and approximate it as a normal distribution. Initially, it has a normal distribution because it comes from the Kalman filter of Eq. (7).

$$P_{of}^{ij} = \mathcal{N}(\pi_i, \Sigma_{x_i,0})$$

Because $P_{of}^{ij}$ is a normal distribution, the probability at the next iteration, $P_{of,1}^{ij}$ can be estimated as follows.

$$\tilde{X}_{f,1} \sim \mathcal{N}(A_p X_{0}, A_p \Sigma_{X,0} A_p^T + \Sigma_w)$$

If we assume that all the subsequent probabilities are also normal distributions as shown in Appendix B-B, any $\tilde{X}_{f,k}$ can be predicted from its previous estimate as follows.

$$\tilde{X}_{f,k}^{ij} \sim \mathcal{N}(A_p \tilde{X}_{f,k-1}^{ij}, A_p \Sigma_{X_{f,k-1}}^{ij} A_p^T + \Sigma_w)$$

REFERENCES

[1] F. Aghili. A prediction and motion-planning scheme for visually guided robotic capturing of free-floating tumbling objects with uncertain dynamics. IEEE Transactions on Robotics, 28(3):634–649, 2012.
[2] D. Berenson, S. S Srinivasa, D. Ferguson, and J. J. Kuffner. Manipulation planning on constraint manifolds. In *Robotics and Automation, 2009. ICRA’09. IEEE International Conference on*, pages 625–632. IEEE, 2009.

[3] S. Bouraine, T. Fraichard, and H. Salhi. Relaxing the inevitable collision state concept to address provably safe mobile robot navigation with limited field-of-views in unknown dynamic environments. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2985–2991. IEEE, 2011.

[4] A. Cherubini, F. Spindler, and F. Chaumette. Autonomous visual navigation and laser-based moving obstacle avoidance. *IEEE Transactions on Intelligent Transportation Systems, 15(5):2101–2110, 2014.

[5] N. D. Tuò and J. W. Burdick. Probabilistic collision checking with chance constraints. *IEEE Transactions on Robotics, 27(4):809–815, 2011.

[6] T. Fraichard and H. Asama. Inevitable collision statesa step towards safer robots? *Advanced Robotics, 18(10):1001–1024, 2004.

[7] S. Haddadin, A. Albu-Schäffer, and G. Hirzinger. Requirements for safe robots: Measurements, analysis and new insights. *The International Journal of Robotics Research, 28(11-12):1507–1527, 2009.

[8] S. Haddadin, A. De Luca, and A. Albu-Schäffer. Robot collisions: A survey on detection, isolation, and identifications. *IEEE Transactions on Robotics, 33(6):1292–1312, 2017.

[9] K. Hauser, V. Ng-Thow-Hing, and H. Gonzalez-Baños. Multi-modal motion planning for a humanoid robot manipulation task. In *Robotics Research, pages 307–317. Springer, 2010.

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

[11] B. Kim, T. T. Um, C. Suh, and F. C. Park. Tangent bundle rrt: A randomized algorithm for constrained motion planning. *Robotica, 34(01):202–225, 2016.

[12] K. S. Kim, A. S. Kwo, and L. Senti. Contact sensing and mobility in rough and cluttered environments. In *Mobile Robots (ECMR), 2013 European Conference on*, pages 274–281. IEEE, 2013.

[13] K. S. Kim, T. Llado, and L. Sentis. Full-body collision detection and reaction with omnidirectional mobile platforms: A step towards safe human-robot interaction. *Autonomous Robots, 40(2):325–341, 2016.

[14] S. Kim and A. Billard. Estimating the non-linear dynamics of free-flying objects. *Robotics and Autonomous Systems, 60(9):1108–1122, 2012.

[15] J. J. Kuffner and S. M. LaValle. Space-filling trees: A new perspective on incremental search for motion planning. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on*, pages 2199–2206. IEEE, 2011.

[16] A. Lambert, D. Gruyer, and G. Saint Pierre. A fast monte carlo algorithm for collision probability estimation. In *Control, Automation, Robotics and Vision, 2008. ICARCV 2008. 10th International Conference on*, pages 406–411. IEEE, 2008.

[17] M. Levihn, M. Stilman, and H. Christensen. Locally optimal navigation among movable obstacles in unknown environments. In *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pages 86–91. IEEE, 2014.

[18] Z. Lin, V. Zeman, and R. V. Patel. On-line robot trajectory planning for catching a moving object. In *Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on*, pages 1726–1731. IEEE, 1989.

[19] B. Léspérance and K. Gupta. Safety hierarchy for planning with time constraints in unknown dynamic environments. *IEEE Transactions on Robotics, 30(6):1398–1411, 2014.

[20] T. McMahon, S. Thomas, and N. M. Amato. Sampling-based motion planning with reachable volumes: Theoretical foundations. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 6514–6521. IEEE, 2014.

[21] D. J. Meagher. Octree encoding: A new technique for the representation, manipulation and display of arbitrary 3-d objects by computer. *Electrical and Systems Engineering Department Rensseiaer Polytechnic Institute Image Processing Laboratory, 1980.

[22] M. Mehrandezh, N. M. Sela, R. G. Fenton, and B. Benhabib. Robotic interception of moving objects using an augmented ideal proportional navigation guidance technique. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 30(3):238–250, 2000.

[23] R. Munoz-Salinas. Aruco: a minimal library for augmented reality applications based on opencv. *Universidad de Cordoba, 2012.

[24] Aslam Pervez and Jeha Ryu. Safe physical human robot interaction-past, present and future. *Journal of Mechanical Science and Technology, 22(3):469, 2008.

[25] L. Sentis, J. Petersen, and R. Philippsen. Experiments with balancing on irregular terrains using the dreamer mobile humanoid robot. In *Robotics: Science and Systems, 2012.

[26] J. Sheil and I. O’ Muircheartaigh. Algorithm as 106: The distribution of non-negative quadratic forms in normal variables. *Journal of the Royal Statistical Society. Series C (Applied Statistics), 26(1):92–98, 1977.

[27] I. Suman and E. E. Kavrakli. A sampling-based tree planner for systems with complex dynamics. *IEEE Transactions on Robotics, 28(1):116–131, 2012.

[28] M. Vasic and A. Billard. Safety issues in human-robot interactions. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 197–204. IEEE, 2013.

[29] Y.-H. Weng, C.-H. Chen, and C.-T. Sun. Toward the human–robot co-existence society: on safety intelligence for next generation robots. *International Journal of Social Robotics, 1(4):267–282, 2009.

[30] Z. Yao and K. Gupta. Path planning with general end-effector constraints: Using task space to guide configuration space search. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 1875–1880. IEEE, 2005.