Contact Localization using Velocity Constraints

Sean Wang[1], Ankit Bhatia[2], Matthew T. Mason[2], and Aaron M. Johnson[1]

***Abstract*— Localizing contacts and collisions is an important aspect of failure detection and recovery for robots and can aid perception and exploration of the environment. Contrary to state-of-the-art methods that rely on forces and torques measured on the robot, this paper proposes a kinematic method for proprioceptive contact localization on compliant robots using velocity measurements. The method is validated on two planar robots, the quadrupedal Minitaur and the two-fingered Direct Drive (DD) Hand which are compliant due to inherent transparency from direct drive actuation. Comparisons to other state-of-the-art proprioceptive methods are shown in simulation. Preliminary results on further extensions to complex geometry (through numerical methods) and spatial robots (with a particle filter) are discussed.**

I. Introduction

Robots are proving to be increasingly useful in unstructured environments, such as cluttered homes and outdoor terrain. However, in these environments robots must deal extensively with the making and breaking of contact with uncertain or unknown object shapes and poses. This makes contact localization a vital skill. For example, Fig. 1 shows a legged robot walking up stairs, with one of its legs in contact with the edge of a step. In this scenario, noisy estimation of the height of a stair can lead to unexpected contact between the edge and the leg causing the robot to trip. Another example is an industrial robot gripper with planar fingers approaching an object to grasp or estimate its surface, Fig. 9.

In this paper, we propose a generalized extension to the method used by Barasuol *et. al.* [1] for planar velocity-based contact localization, summarized in Fig. 1. The method is based on the observation that if a point is in contact with a rigid body, its velocity in the direction of the surface normal must be zero, i.e., its velocity is perpendicular to the surface normal. Calculating the instantaneous velocity at each point on the surface will yield a set of candidate points.

The main requirement for this method to work is that the robot be rigid and have accurate position and velocity measurements on the collision link. Any motion due to collision is directly transmitted to the link's position/velocity sensors. The robots used for this work are transparent in the conversion of external force to motion due to their direct drive joints. That is, any external forces are efficiently relayed to motion at the joints that can be picked up by joint encoders. Adding mechanical or software compliance to the articulated joints are alternate ways to achieve high transparency. Adding inertial measurement units can also provide the information needed to compute collisions with the velocity contact localization method.

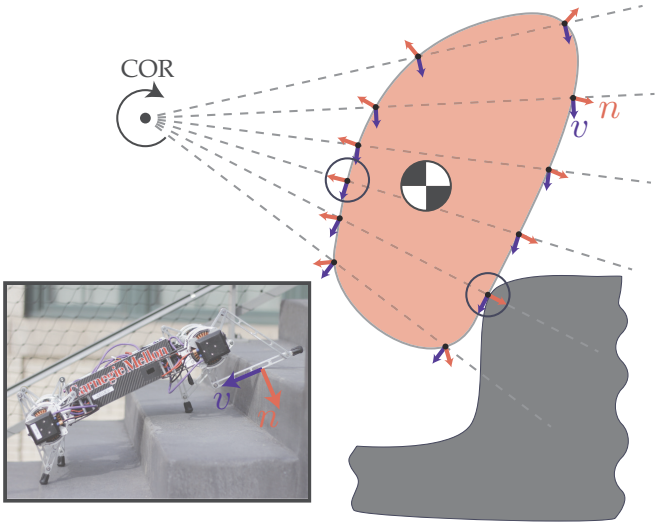

Fig. 1: A velocity-based contact localization scheme. Contact is localized to a set of candidate points (circled) where the surface normal n is co-linear with the line joining the candidate point to the center of rotation (COR). Equivalently, the velocity at the candidate point v must be perpendicular to the surface normal. Inset: Example application of a quadruped robot on a flight of stairs.

We analyse this method and claim that it has the following properties: 1) the method provides an instantaneous estimate of contact point locations. 2) The method does not require an accurate dynamical model of the robot. 3) The method can be implemented on existing robots without any additional sensors beyond joint position or velocity measurements. 4) The method uses a velocity constraint to produce a set of possible contact points. For general planar systems, including the legged robot and gripper of interest to us, this is sufficient to isolate the contact point to a 0-dimensional set (i.e. an individual point or, if there is ambiguity in shape, a set of possible points). For spatial systems, this produces a 1-dimensional set of possible contact points (one or multiple curves). In general, it produces a codimension 1 set of points, i.e., an n-dimensional surface in collision should produce an n-1 dimensional set of possibilities. This can be reduced down to a single contact point through filtering, assuming frictional contact, or adding additional dynamic constraints.

We evaluate the performance of the velocity-based method and compare it with position and torque-based methods in a

*This work was supported in part by a National Science Foundation Grant (IIS-1813920) and by Carnegie Mellon University's Presidential Fellowship.

The first two authors contributed equally to this work.

[1]Department of Mechanical Engineering, Carnegie Mellon University, Pittsburgh, PA, USA

[2]Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

simulation of a five-link planar robot. We also demonstrate the velocity based method in hardware experiments with a legged robot (Minitaur [2]) and a two-finger gripper (DD Hand [3]). We then show a proof of concept to extend the planar method to 3D with the help of a particle filter.

A. Related Work

It is common to localize unknown contacts with exteroceptive sensors such as LIDAR, structured light, and stereo cameras. However, these sensors often do not give the accuracy needed to perform precise tasks such as fine manipulation or maneuvering in a tight space. Furthermore, in cases with poor lighting, feature starvation, or occlusion, these sensors may fail to detect obstacles, leaving the robot unable to anticipate impending collisions. This drives the need for fast and robust contact sensing and localization.

One popular method to achieve this employs tactile sensors, e.g. [4,5]. A sensitive skin is applied to the surface of the robot that can then measure contact forces and/or locations. Unfortunately, the implementation of tactile sensors leads to undesired design constraints, higher cost, and limited material choices. In fact, most robots utilize sensorized skin only on areas with high probability of contact, such as fingertips and feet, leaving the rest of their body uncovered (with some notable exceptions, e.g. [6–8]). Tactile sensors are also limited in their sensitivity and spatial resolution.

Methods based on proprioceptive sensing can overcome some limitations of local tactile sensing. Proprioceptive sensors measure the internal state of the robot (joint positions, velocities, and torques) and can ideally detect contact anywhere on the robot structure. Fortunately, most robots already measure their internal state for control using joint encoders and torque sensors so no additional hardware is required. Position, velocity, and torque signals can all be used to localize contacts. Torque-based contact localization methods are the most popular [9,10]. However, a major drawback is that they require accurate dynamics to estimate external forces. Uncertainty in parameters such as weight distribution, friction, and damping limit the accuracy of these methods.

For a suitable subset of robot problems, just position [11,12] or velocity [1] can be sufficient to localize contact. Position-based methods require position measurements to be spaced out in time and assume stationary point contacts. This makes them unfavorable for moving contacts such as when the contact is rolling or sliding. Velocity-based contact localization methods are less common, but recent work on the HyQ robot [1] suggests they hold merit.

II. Problem Setup & Notation

The paper follows modified notations from [13]. All vectors are denoted with lowercase boldface fonts (e.g. $\mathbf{a}$ or $\mathbf{c}$), all matrices are uppercase boldface font (e.g. $\mathbf{J}$ or $\mathbf{Ad}$), all sets are written in uppercase font (e.g. B or C), and all frames are written in roman typestyle (e.g. P or L.) The terms defined in this section are shown in Fig. 2.

We consider a robot with m actuators and n rigid links. We assume that the robot motion is constrained to be planar (although we relax this assumption in Section VI-B). We assume that the robot has an accurate knowledge of its geometry, and can measure its actuator positions and velocities, which are denoted as $\mathbf{q} = (q_1, \ldots, q_m)^T$ and $\dot{\mathbf{q}} = (\dot{q_1}, \ldots, \dot{q_m})^T$.

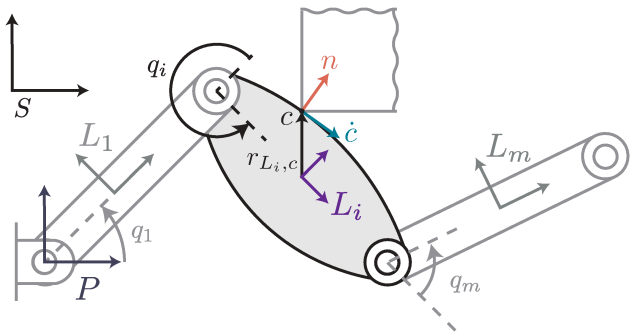

Fig. 2: Schematic of a generic robot with m actuators showing the notation used in this paper.

We also assume that the robot has an accurate estimate of its body velocity twist $\mathbf{v}^b_{\mathrm{S,P}}$, where S is a frame fixed to the world, and P is a frame fixed to the robot's body. We define n frames $\mathrm{L}_1, \mathrm{L}_2, ..., \mathrm{L}_n$ that are fixed to each link of the robot (such that the transformation between P and L_i is described entirely by $\mathbf{q}$). At time t_0 the i^{th} link of the robot collides with a stationary foreign object and is sliding along the object until time $t_f > t_0$. We assume that the robot is compliant enough that it does not come to a complete rest after colliding with the object, but continues to roll or slide along the object ($\|\dot{\mathbf{q}}(t)\| + \|\mathbf{v}^b_{\mathrm{S,P}}\| \neq 0 \ \forall t \in [t_0, t_f]$). We define B as the set of all points on the surface of link i at which contact could have occurred. Note that these could be on any surface of the link. It is also possible to pre-filter this set for surface points that are not expected to be in contact. For a point $\mathbf{c} \in B$, the vector $\mathbf{r}_{\mathrm{L}_i,\mathbf{c}}$ is defined as the coordinates of point $\mathbf{c}$ in the L_i frame. The vector $\mathbf{n}_{\mathrm{L}_i,\mathbf{c}}$ is defined as the unit vector normal to link i's surface and pointed outwards at point $\mathbf{c}$, also in the L_i frame.

We define $\mathbf{J}_i$ as the body Jacobian that maps actuator velocities, $\dot{\mathbf{q}}$, to the body velocity twist $\mathbf{v}^b_{\mathrm{P,L}_i}$ of frame L_i relative to frame P,

$$\mathbf{v}^b_{P,L_i} = \mathbf{J}_i \dot{\mathbf{q}}. \tag{1}$$

$\mathbf{J}_i$ only explains the internal motion of the frame L_i in the robot's base frame. To account for it's spatial motion, we also need to consider $\mathbf{v}^b_{\mathrm{S,P}}$.

Given the homogeneous transformation matrix $\mathbf{H}_{1,2}$ between two frames 1 and 2, with rotation matrix $\mathbf{R}_{1,2}$ and translation vector $\mathbf{r}_{1,2}$, define the adjoint matrix $\mathbf{Ad}_{\mathbf{H}_{1,2}}$ as,

$$\mathbf{Ad}_{\mathbf{H}_{1,2}} = \begin{bmatrix} \mathbf{R}_{1,2} & \hat{\mathbf{r}}_{1,2}\mathbf{R}_{1,2} \\ 0 & \mathbf{R}_{1,2} \end{bmatrix}, \tag{2}$$

where $\hat{\mathbf{r}}_{1,2}$ is the skew symmetric matrix of vector $\mathbf{r}_{1,2}$. For more information on the adjoint operator in screw theory, the reader is referred to [13].

Upon contact with the foreign object, the robot must localize the point of contact, that is, compute $\mathbf{r}_{\mathrm{L}_i,\mathbf{c}}$.

III. Review of proprioceptive methods

In this paper, we compare our method of contact localization against two other proprioceptive methods. A brief review of these methods follows.

A. Position-based Contact Localization

Position-based methods depend on self-posture changing motions [11]. The robot is assumed to have collided with a stationary wedged-shaped object. Hence, the point of contact remains stationary in the world frame S. To estimate the location of contact, joint positions at two instances $[t-\Delta, t]$ during the contact time window $[t_0, t_f]$ are recorded. The method is generally accompanied by an exploratory motion to space out the two measurements. The intersection of the surface of link i at these positions localizes the contact.

If the object breaks the assumption that it is a wedge (a curved object, for example) the estimated contact location will have some error and may be off of the object as the contact location will have moved between the two measurements. To improve the accuracy for such cases, the measurements can be moved closer together in time which will approach the proposed velocity-based method. Note that as the measurements become closer, the estimated contact location will become increasingly noisy.

B. Torque-based Contact Localization

This method, also known as intrinsic tactile sensing [14], assumes that an unknown linear force, $\mathbf{f}_{ext}$, and zero moment is applied at the contact point due to collision with the foreign object. This results in the wrench, $[\mathbf{f}_{ext}^T, \mathbf{0}^T]^T$, in frame L_i, applied at the point $\mathbf{c}$. Applying this wrench at the point $\mathbf{c}$ is equivalent to applying the wrench $\mathbf{f}_i$ at frame L_i, where $\mathbf{f}_i$ is given by,

$$\mathbf{f}_i = \begin{bmatrix} \mathbf{f}_{ext} \\ \mathbf{r}_{\mathrm{L}_i,\mathbf{c}} \times \mathbf{f}_{ext} \end{bmatrix}. \tag{3}$$

It is also equivalent to an external torque, $\boldsymbol{\tau}_{ext} \in \mathbb{R}^m$ applied to the robot's actuators,

$$\boldsymbol{\tau}_{ext} = \mathbf{J}_i^T \mathbf{f}_i. \tag{4}$$

With a momentum observer, velocity observer, or another collision monitoring method (as in [15]), the external torque, $\boldsymbol{\tau}_{ext}$ can be estimated.

For this method, it is required that the robot can measure these actuator torques for which an accurate dynamical model of the robot might be required. With the estimate of the external torques $\boldsymbol{\tau}_{ext}$, equations (3) and (4) can be solved for $\mathbf{r}_{\mathrm{L}_i,\mathbf{c}}$, and $\mathbf{f}_{ext}$.

For a robot in the plane, the surface of link i is one dimensional. Since the contact point lies on this one dimensional surface, $\mathbf{r}_{\mathrm{L}_i,\mathbf{c}}$ can be reduced to one unknown variable. As $\mathbf{f}_{ext}$ has unknown magnitude and direction, $\boldsymbol{\tau}_{ext}$ needs to be at least 3 elements long for the problem to have a unique solution. This implies the robot must have at least three degrees of freedom (DOF). If a further assumption of frictionless contact can be made, $\mathbf{f}_{ext}$ can be reduced to one unknown variable and this method can generate a unique solution on a 2 DOF robot. Similarly a spatial robot must have at least 5 DOF, or 3 DOF in the frictionless case, to generate a unique solution.

IV. Proposed Velocity-based Method

The torque-based method uses the transpose of the Jacobian to relate end-effector force to joint torques. Here, we propose a method that utilises its dual relationship: use the Jacobian to map joint velocities to end-effector velocity. That, combined with the velocity constraint enforced by a collision, can localize the contact point.

We begin by summarizing the assumptions that need to hold to use velocity for accurate contact localization.

1) The robot must be compliant and have at least one degree of freedom after applying the constraint. That implies that the robot does not come to a complete stop after collision.
2) Contact detection is solved, i.e., we know the time of collision and which link has collided with the external constraint.
3) A good estimate of body position and velocity of the collision link is available. The better this estimate, the more accurate the contact localization.

To derive the location of the contact point, we need to find the linear velocity of a point $\mathbf{c}$ on the surface of the link L_i. Let us first express the body velocity twist of frame L_i in terms of the generalized coordinates,

$$\mathbf{v}_{S,L_i}^b = \mathbf{Ad}_{\mathbf{H}_{\mathrm{P},\mathrm{L}_i}^{-1}} \mathbf{v}_{\mathrm{S,P}}^b + \mathbf{J}_i \dot{\mathbf{q}}. \tag{5}$$

With this, we can express the linear velocity of point $\mathbf{c}$, in terms of the velocity of frame L_i at time t as,

$$\dot{\mathbf{c}}(t) = \begin{bmatrix} I & -\hat{\mathbf{r}}_{\mathrm{L}_i,\mathbf{c}} \end{bmatrix} \mathbf{v}_{\mathrm{S},\mathrm{L}_i}^b(t). \tag{6}$$

Based on this, the scalar velocity in the normal direction, $\dot{c}_n(t)$, of a point $\mathbf{c}$ at time t is,

$$\dot{c}_n(t) = \mathbf{n}_{\mathrm{L}_i,\mathbf{c}} \cdot \dot{\mathbf{c}}(t), \tag{7}$$

where $\mathbf{n}_{\mathrm{L}_i,\mathbf{c}}$ is the surface normal of link L_i at $\mathbf{c}$.

A. The Method

There are two velocity constraints that must hold at the true contact point ($\mathbf{c}^*$). First, during contact, $\mathbf{c}^*$ must have zero velocity in the direction normal to the link surface,

$$\dot{c}_n^*(t) = 0 \quad \forall t \in [t_0, t_f]. \tag{8}$$

This is necessary for persistent contact between the link and the object. For the planar case, this constraint is equivalent to having the line from the center of rotation to the point $\mathbf{c}^*$ be perpendicular to the robot's surface, as shown in Fig. 1.

Second, at the instant before contact, denoted as t_0^-, the point of initial contact must have a positive velocity along the surface normal of the link,

$$\dot{c}_n^*(t_0^-) > 0. \tag{9}$$

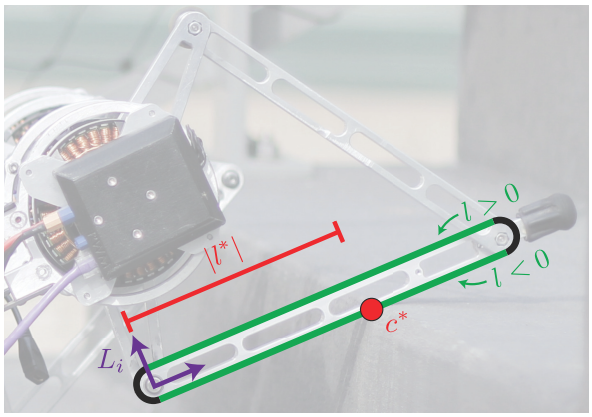

Fig. 3: The portion of the link that is expected to make contact (B) is shown in green. Any point in this one dimensional set can be characterized by the parameter l. Using the velocity-based method, the robot can find l^*, which characterizes the true contact point.

Thus, to localize a contact, the proposed method simply identifies the set of possible contact points, denoted C, which contains candidate points that satisfy both constraints. An example of computing the set C for a case with simple link geometry is given below. An algorithm to compute C when there is complex link geometry is shown in Section VI-A.

The codimension of C in B is one. For the planar case, since the set B is one dimensional, the set C is zero dimensional. In many planar cases, C will contain a unique possible contact point at time t_0. For a spatial robot, the set B is two dimensional, so the set C is one dimensional. Reducing contact location ambiguity in cases where C contains more than one point is discussed in Section VI-B.

B. Simple Geometry Example

We now apply this method to an example robot with simple link geometry. Consider a Minitaur robot that has collided with the edge of a stair, Fig. 1. Using one of the methods described in [15], the robot has detected that the collision occurred at time t_0 and lasts until time t_f. Fig. 3 shows the link of the robot that has made contact. The portion of this link's surface that is expected to make contact, B, is highlighted in green. Any arbitrary point $\mathbf{c}$ on this highlighted region can be characterized by the variable $l \in \mathbb{R}$, where $|l|$ is the distance away from the frame L_i, and $sgn(l)$ denotes which side of the link the point is on. If d is the width of the link, then

$$\mathbf{c} = \begin{bmatrix} |l| & \frac{d}{2} sgn(l) \end{bmatrix}^T. \tag{10}$$

The normal velocity of the point $\mathbf{c}(l)$, can be written explicitly as a function of l. If $\mathbf{v}^b_{\mathrm{S,L}_i}(t) = [v_x, v_y, \omega_z]^T$, then

$$\dot{c}_n(t) = sgn(l)(v_y + |l|\omega_z). \tag{11}$$

At time t ($t_0 > t > t_f$), the values of l that satisfies the constraint in (8) is,

$$\dot{c}_n(t) = 0 \Rightarrow l = \pm v_y/\omega_z \tag{12}$$

The set of possible contact points, C can then be found by mapping these values of l to points using (10).

For $t_0 < t < t_f$, the set C contains two possible contact points. However, at time t_0, the candidate contact points also need to satisfy $\dot{c}_n(t_0^-) > 0$, due to the constraint in (9). Given that $l = \pm v_y/\omega_z$ for the two points in C and (11), one of the points in C violates the constraint in (9) and is removed from C. This results in a unique point at which contact could have occurred.

V. Implementation Results

A. Contact Localization in Simulation

In ideal conditions, the position, velocity, and torque based solutions all provide accurate contact localization. However, they differ in their sensitivity to noise. To evaluate this, we simulate the frictionless collision of a five-bar linkage with a point constraint. We assume that the collision link is fixed and known. We derive the kinematics and differential kinematics of the five-bar linkage as described in [16] and the dynamics using a constrained Lagrangian approach [13]. The dynamics are then projected onto the reduced coordinate space of the actuated joints. A constraint is added to this system for a frictionless point contact. This system is simulated in an event-driven framework using `ode45` in MATLAB. The linkage is actuated with a Proportional-Derivative position controller computed at 500 Hz outputting joint torques. The parameters for the simulation are shown in Table I. A trace of the simulation is shown in Fig. 5 with the start configuration (green) and the commanded configuration (dashed black). The constraint and the predicted contact location under ideal conditions are shown in red.

Using this simulation environment, we can test the sensitivity of the three algorithms described in this paper to injected noise in the process parameters. The velocity and torque methods were tested twice: once with only encoder position as input and again with both encoder position and velocity as input. For the first case, labeled "Velocity($\mathbf{q}$)" and "Torque($\mathbf{q}$)", the velocity was computed by finite difference of the encoder position at the same dt (0.02 s) as the time window of the position-based method. In this case, the velocity method was indiscernible from the position method and are shown together. For the second case, labeled "Velocity($\mathbf{q}, \dot{\mathbf{q}}$)", and "Torque($\mathbf{q}, \dot{\mathbf{q}}$)", a velocity was computed at a higher frequency (4000 Hz) and then low pass filtered (3 dB cutoff frequency of 142 Hz.) The simulation executed nominally with a 16 bit encoder resolution as baseline noise.

Parameter	Value
Link Masses (m_1,m_2,m_3,m_4)	0.1 kg
Link Lengths (l_1, l_2, l_3, l_4, l_5)	0.08,0.15,0.15,0.08,0 m
Gravity	9.81 m s^{-2}
Contact Location	0.05 m, 0.09 m
Control Frequency	500 Hz
Position time window	0.02 s
Proportional Gain K_p	0.5 kg s^{-2}
Derivative Gain K_d	0.04 kg s^{-1}

TABLE I: Parameters used for simulation experiments.

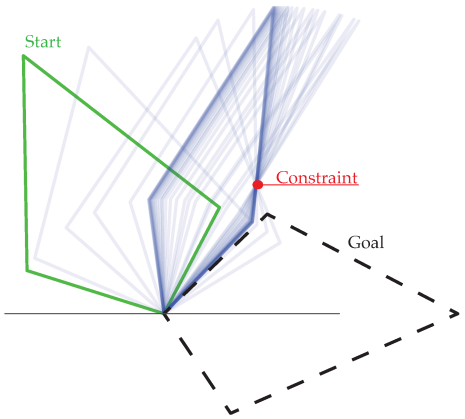

Fig. 5: A sample trace of the simulation of the collision of a five-bar mechanism (one link is of length zero) with a point constraint(red) used to gauge the sensitivity of the discussed algorithms to model and process noise. The green trace is the initial state of the linkage which is commanded to align it's major axis with the x-axis. The darker blue trace is the resting position of the linkage after contact.

For all sensitivity analyses, the performance is evaluated using the error over a $100\,\mathrm{ms}$ window starting $100\,\mathrm{ms}$ after contact is detected. Root-Mean-Squared (RMS) error over 100 trials of each condition is reported in Fig. 4. The results described below are all statistically significant with a p value of less than 0.05. The z-statistic was used to compare the distribution of error computed at the extremes of the parameter sweep.

1) Sensitivity to Encoder noise: Gaussian noise was added to the generalized position coordinates from the simulation to emulate encoders with 8 to 20 bits of resolution with the standard deviation as 1 encoder count in radians. This proved to be the most important parameter tested, with all methods showing a large increase in RMS error with decreasing encoder resolution (Fig. 4a). The velocity-method with position and velocity input, Velocity($\mathbf{q}, \dot{\mathbf{q}}$), was the best method at low to moderate encoder resolutions, while all methods had sub-mm error for higher resolutions.

2) Sensitivity to Torque Noise: The torque based contact localization method requires the measurement of the generalized torques (joint torques in this case) to compute the external torques. Uniform torque noise of 0 to $0.2\,\mathrm{N\,m}$ was added to the measured torque. The torque method with position and velocity input, Torque($\mathbf{q}, \dot{\mathbf{q}}$), shows a significant increase in RMS error with increase in torque noise (while Torque($\mathbf{q}$) had uniformly higher error).

3) Sensitivity to Geometric Noise: Noise in measured dimensions are possible due to errors in design and manufacturing. We analyze the effect of up to 20% noise in the geometric parameters of the linkage. All methods except Torque($\mathbf{q}$) showed a statistically significant rise in error with increasing geometric noise, although the magnitude of the increase was less than for the other parameters tested. We attribute this to the weak signal to noise output of this method due too its dependence on a clean velocity signal.

4) Sensitivity to Inertial Noise: Careful calibration of geometric and inertial properties of each link and its contribution to the dynamic response can be tedious [17,18]. To analyze the effect of errors in the dynamic robot model, we simulate up to 100 percent noise in the mass parameters of the linkage. As expected, both torque-based methods showed a statistically significant rise in error.

B. Contact Localization on the Minitaur Leg

The three contact localization methods were implemented and compared on a Minitaur robot. During contact localization, motor positions were measured using encoders mounted on the motors and velocities were estimated by taking the first order numerical differentiation of motor position. Motor torques were estimated from measured motor currents. For all three methods, it was assumed that contact occurred on the last link of the robot's leg.

A $100\,\mathrm{ms}$ time window was chosen for both the time window of the position-based method and the dt of the velocity estimate. The effects of varying this time window and the dt value are discussed below.

As discussed in Section III-B, the torque-based method is dependent on an accurate dynamic model of the robot. This model was derived from link and motor mass distributions

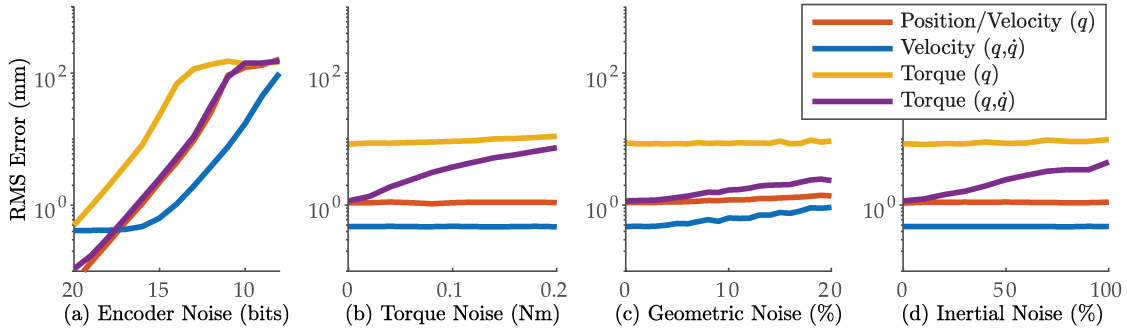

Fig. 4: Effect of process and parametric noise on the accuracy of contact localization methods in simulation. All methods are highly sensitive to changes in position accuracy. The position and velocity contact localization outperform torque-based under all perturbations.

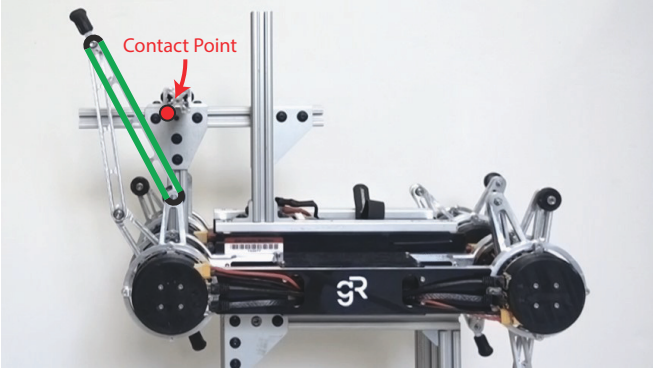

Fig. 6: An experiment localizing contact on a Minitaur robot. A video of the experiment is attached.

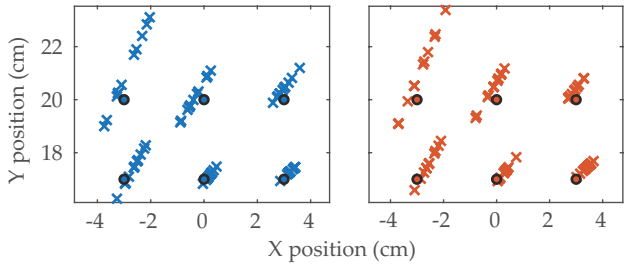

Fig. 7: Estimated contact locations from the Minitaur experiment using the velocity-based method (left) and the position-based (right). The circles represent the actual contact locations, and the crosses represent the estimations made using either method. With the same time window, both methods result in similar contact location estimates.

as measured from CAD models and actual link weights. Aerodynamic effects and frictional forces were ignored. Motor constants vary between individual motors and as such they were calibrated separately for each motor. Furthermore, as mentioned in Section III-B, it is necessary to make a frictionless contact assumption to find a single solution.

Unfortunately, we were unable to obtain sensible results using the torque-based method. While the estimated external torque values (τ_{ext}) from the momentum-observer seemed to be realistic, the contact locations were not. We believe that this is due to the high sensitivity of contact location to any noise in the τ_{ext} estimates. This difficulty in achieving an accurate torque-based estimate provides further motivation for the method developed in this work.

To compare the velocity-based method and position-based method, one of the robot's legs was swept into a stationary object and the two methods were used to estimate the contact locations. These estimated contact locations were then compared to ground truth contact location measurements. To get these accurate ground truth measurements, the obstacle was rigidly attached to the body of the robot, and the dimensions of the rig connecting the obstacle to the robot were measured. This experiment provides a scenario similar to a legged robot on stairs (Fig. 1), but provides ground truth contact locations, which was used to evaluate contact location estimates, within $1\,\mathrm{cm}$ of the true contact location. The experiment setup is shown if Fig. 6. Six different contact positions were used. At each position, 100 estimations of the contact location were made using the three different methods. The actual contact locations are shown as the dots in Fig. 7 and the estimated contact locations from the position-based, and velocity-based methods are shown as the crosses.

With a $100\,\mathrm{ms}$ time window for the position-based method and dt for the velocity estimate, the accuracy of the position-based and velocity-based methods was comparable. The position-based method had an average error of $0.57\,\mathrm{cm}$, with a variance of $0.14\,\mathrm{cm}$. The velocity-based method had an average error of $0.48\,\mathrm{cm}$, with a variance of $0.12\,\mathrm{cm}$.

The time window of the position-based method and the dt of the velocity estimates were simultaneously varied from $10\,\mathrm{ms}$ to $100\,\mathrm{ms}$ to analyse their effects on the contact

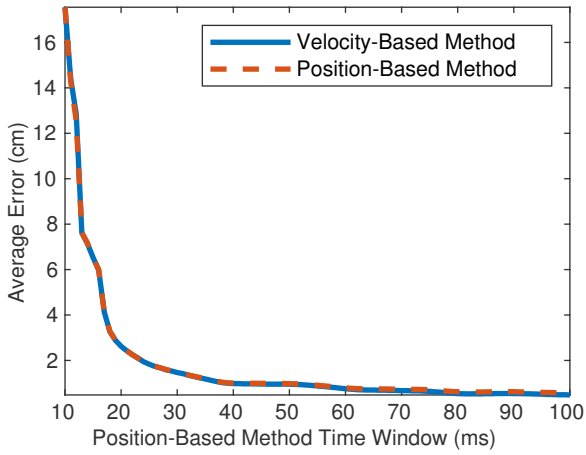

Fig. 8: Average error of estimated contact point position from the Minitaur experiment using both the position-based method and the velocity-based method with a varying time window and velocity estimate dt size.

location estimates. The results of this experiment is shown in Fig. 8. As expected, with a shorter time window and dt, both method resulted in higher error. Throughout the range of tested time windows and dt values, both methods of contact localization had indistinguishable error. However, as shown in Section V-A, the velocity-based method could have performed better than the position-based method if other methods of velocity estimation were used.

C. Contact Localization on the DD Hand

The DD Hand [3] is a direct drive gripper with two five-bar linkages as fingers. To demonstrate the velocity-based method with a non-point contact, we collide one of the fingers with a cylinder and continuously track the estimated contact location. Fig. 9 shows the evolution of the linkage during the experiment along with the estimated contact points overlaid. The estimated contact location is seen tracking the curved surface of the cylinder. The small deviation at the start can be attributed to the lag due to the velocity filter running on board the DD Hand.

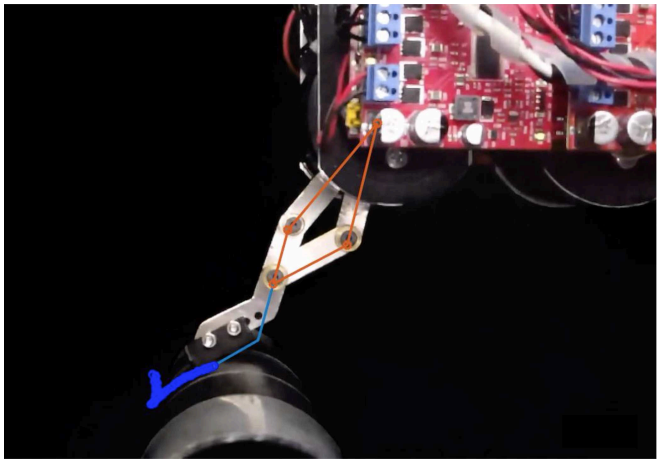

Fig. 9: A snapshot from the experiment to detect contact of the DDHand with a cylindrical object. A video of the experiment is attached.

Algorithm 1: Numerical Contact Localization

$\bar{B} \leftarrow$ discretized surface
$C \leftarrow \emptyset$
for $\mathbf{c} \in \bar{B}$ **do**
  if $|\dot{c}_n(t)| < \epsilon$ **then**
    $C = C \cup \mathbf{c}$
  end
end
if $t = t_0$ **then**
  for $\mathbf{c} \in C$ **do**
    if $\dot{c}_n(t_0^-) < 0$ **then**
      $C = C \setminus \mathbf{c}$
    end
  end
end

VI. Extending Velocity Contact Localization

In this section, we present preliminary extensions to the method discussed in this paper to robots with non-trivial geometries and robots with 2D link surfaces moving in space.

A. Numerical Methods for Complex Geometries

For links with complex surface geometry, using a similar approach to section IV to solve for C explicitly may be hard. Algorithm 1 can be used to numerically solve for C. In this algorithm, the continuous set B is discretized into the set $\bar{B} = \{\mathbf{c}_1, \mathbf{c}_2, ..., \mathbf{c}_k\}$. This discretization must be done in a fashion such that the set $\bar{B}$ provides good spatial coverage of the set B. Then at time t, $\dot{c}_n(t)$ is evaluated for every point $\mathbf{c}$ in $\bar{B}$. If the magnitude of $\dot{c}_n(t)$ is under a set threshold ϵ, then the point is added to the set C. Since the initial contact point, c_0^*, also needs to satisfy the constraint in (9), at time $t = t_0$, points in C are additionally checked for compliance to the constraint $\dot{c}_n(t_0^-) < 0$ and removed if in violation.

B. Particle Filtering for Spatial Contact Localization

As mentioned in Section IV, the velocity-based method may not always return a unique point possible contact point. This could be due to either ambiguities in linkage geometry or the nature of spatial motions. For example, Fig. 10 shows a cylindrical robot link impacting the edge of a box. Using the proposed velocity-based method at the time of impact ($0\,\mathrm{ms}$) leads to uncertainty about the contact point location. The black line in the top left subfigure in Fig. 10 shows the set of possible contact points according to the velocity-based method. The position-based contact localization method also suffers from the same problem, and results in a similar set of possible contact points. In such situations, a particle filter is a useful tool to collapse uncertainty [9,19].

We propose a particle filtering method that uses a motion model that assumes stationary contact and a measurement model that weighs particles based on the set of possible contact points calculated using the velocity-based method.

With the motion model, each particle's position is updated by sampling from a normal distribution centered around the particle's previous position with a covariance matrix Σ_{mo},

$$\mathbf{x}_{t+1}^{[j]} \sim N(\mathbf{x}_t^{[j]}, \Sigma_{mo}). \tag{13}$$

This corresponds to a stationary contact point assumption in the world frame. Note that particles are defined in the world frame and not fixed to the robot's link frame.

The measurement model assumes that the true contact point location plus some noise $\eta \sim N(0, \Sigma_{me})$ lies in the set of possible contact points ($\mathbf{c}^* + \eta \in C$). Based on this assumption, the weight of each particle, which characterizes how well the particle matches the measurements and is used for importance resampling, can be defined as,

$$w(\mathbf{x}_t^{[j]}) = \min_{\mathbf{c} \in C} exp\left(-\frac{1}{2}(\mathbf{c} - \mathbf{x}_t^{[j]})^T \Sigma_{me} (\mathbf{c} - \mathbf{x}_t^{[j]})\right). \tag{14}$$

This proposed particle filter was implemented in the simulation of a cylindrical robot link colliding with the edge of an obstacle, Fig. 10. At the time of impact, $t = 0$, the particles give a good representation of the set of possible contact points given by the velocity-based method. As time evolves, the particles start to converge reducing the uncertainty of the contact location. At $50\,\mathrm{ms}$, the particles have converged to a point that is near the actual contact location.

Although the proposed particle filter performs well in this scenario, it may fail in others. In situations where there is sliding or rolling contact, the stationary contact point assumption made by the motion model will not hold and the particle filter will perform poorly. In the future, a better motion model for the contact location could be created to relax this assumption and allow for situation where the contact point may move in the world frame.

VII. Conclusion

In this paper, we propose a velocity-based method of contact localization. Using velocity constraints, this method provides a codimension 1 set of possible contact points. The advantage of this method over previous methods is that

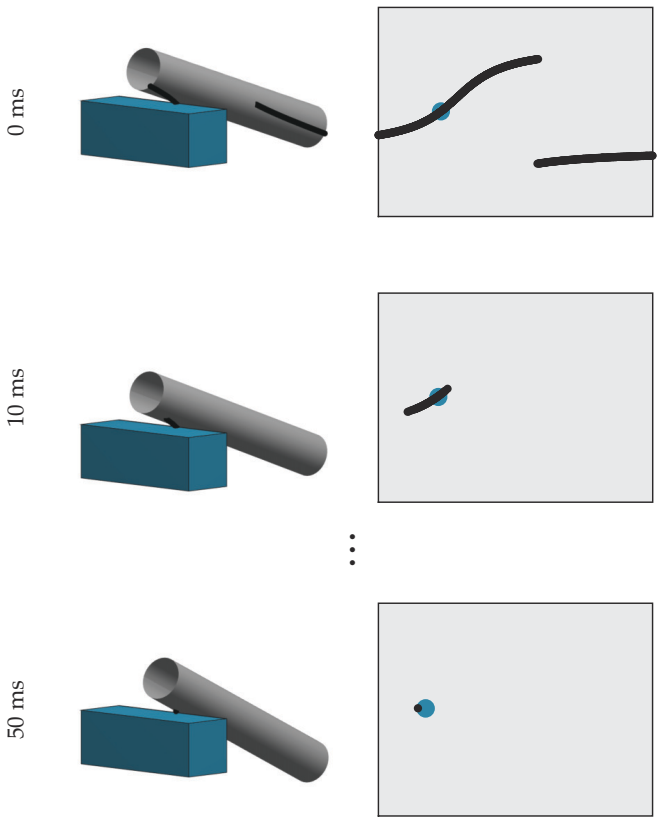

Fig. 10: Left: A cylinder in freefall (grey) interacting with a fixed block(blue). Right: the unrolled surface of the cylinder. A particle filter used to reduce ambiguity of contact location from the velocity-based method. Each particle is shown in black. The actual contact location is shown in blue.

it provides an instantaneous estimate of contact location, does not require a dynamical model of the robot, and only requires robot position and velocity measurements. We validate the performance of this method for planar robots both in simulation and the real world. In the plane, we show that kinematic methods of contact localization are superior to dynamic methods in the presence of noise.

Note that the velocity-based method is equivalent to a position-based method when the velocity is computed as a first order numerical differentiation over the same time window. For small time windows, the velocity and position-based methods produce similar results. This is shown in both the simulation and real-world experiments in Fig. 4 and 8, respectively. However, one advantage of the velocity-based method over the position-based method is that the accuracy of the velocity method improves when a better velocity estimate is provided. This can be achieved with either better numerical differentiation of position data [20] or by fusing other sensor data like acceleration [21].

For spatial robots, this contact localization method results in a one dimensional set of possible contact points leading to ambiguity in the location. To reduce this, we propose the use of a particle filter with a stationary contact point assumption. The capabilities of this particle filter are shown in simulation.

In the future, improvements to state-estimation techniques could allow for more accurate velocity estimates, and thus contact location estimates. Also, development of more intelligent particle filter motion models could extend the method to localizing non-stationary contact points on spatial robots.

References

[1] V. Barasuol, G. Fink, M. Focchi *et al.*, "On the detection and localization of shin collisions and reactive actions in quadraped robots," in *Climbing and Walking Robots*, 2019.

[2] G. Kenneally, A. De, and D. E. Koditschek, "Design principles for a family of direct-drive legged robots," *IEEE Robotics and Automation Letters*, vol. 1, no. 2, pp. 900–907, 2016.

[3] A. Bhatia, A. M. Johnson, and M. T. Mason, "Direct drive hands: Force-motion transparency in gripper design," in *Robotics: Science and Systems*, June 2019.

[4] R. S. Dahiya, P. Mittendorfer, M. Valle *et al.*, "Directions toward effective utilization of tactile skin: A review," *IEEE Sensors Journal*, vol. 13, no. 11, pp. 4121–4138, 2013.

[5] M. Strohmayr, "Artificial skin in robotics," *Doktorarbeit, Karlsruhe Institute of Technology (KIT)*, 2012.

[6] M. Inaba, Y. Hoshino, K. Nagasaka *et al.*, "A full-body tactile sensor suit using electrically conductive fabric and strings," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, vol. 2, 1996, pp. 450–457.

[7] Y. Ohmura and Y. Kuniyoshi, "Humanoid robot which can lift a 30kg box by whole body contact and tactile feedback," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007, pp. 1136–1141.

[8] T. Yoshikai, H. Fukushima, M. Hayashi, and M. Inaba, "Development of soft stretchable knit sensor for humanoids' whole-body tactile sensibility," in *IEEE-RAS International Conference on Humanoid Robots*, 2009, pp. 624–631.

[9] L. Manuelli and R. Tedrake, "Localizing external contact using proprioceptive sensors: The contact particle filter," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 5062–5069.

[10] J. Vorndamme, M. Schappler, and S. Haddadin, "Collision detection, isolation and identification for humanoids," in *IEEE International Conference on Robotics and Automation*, 2017, pp. 4754–4761.

[11] M. Kaneko and K. Tanie, "Contact point detection for grasping an unknown object using self-posture changeability," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 3, pp. 355–367, Jun. 1994.

[12] G. S. Koonjul, G. J. Zeglin, and N. S. Pollard, "Measuring contact points from displacements with a compliant, articulated robot hand," in *IEEE International Conference on Robotics and Automation*, 2011, pp. 489–495.

[13] R. Murray, Z. Li, and S. S. Sastry, *A Mathematical Introduction to Robotic Manipulation*. USA: CRC Press, Inc., 1994.

[14] A. Bicchi, J. K. Salisbury, and D. L. Brock, "Contact sensing from force measurements," *The International Journal of Robotics Research*, vol. 12, no. 3, pp. 249–262, 1993.

[15] S. Haddadin, A. De Luca, and A. Albu-Schaffer, "Robot Collisions: A Survey on Detection, Isolation, and Identification," *IEEE Transactions on Robotics*, vol. 33, no. 6, pp. 1292–1312, Dec. 2017.

[16] G. Campion, Qi Wang, and V. Hayward, "The Pantograph Mk-II: a haptic instrument," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 193–198.

[17] B. Karan and M. Vukobratović, "Calibration and accuracy of manipulation robot modelsan overview," *Mechanism and Machine Theory*, vol. 29, no. 3, pp. 479–500, 1994.

[18] D. Kostic, B. De Jager, M. Steinbuch, and R. Hensen, "Modeling and identification for high-performance robot control: An rrr-robotic arm case study," *IEEE Transactions on Control Systems Technology*, vol. 12, no. 6, pp. 904–919, 2004.

[19] M. C. Koval, M. R. Dogar, N. S. Pollard, and S. S. Srinivasa, "Pose estimation for contact manipulation with manifold particle filters," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 4541–4548.

[20] P. S. Carpenter, R. H. Brown, J. A. Heinen, and S. C. Schneider, "On algorithms for velocity estimation using discrete position encoders," in *IEEE Industrial Electronics Conference*, vol. 2, 1995, pp. 844–849.

[21] W.-H. Zhu and T. Lamarche, "Velocity estimation by using position and acceleration sensors," *IEEE Transactions on Industrial Electronics*, vol. 54, no. 5, pp. 2706–2715, 2007.