

Contact Inertial Odometry: Flying on IMU Only

Tomoki Emmei^{1,2}, Thomas Lew², Tara Bartlett², Angel Santamaria-Navarro², David Fan², Rohan Thakker², and Ali-akbar Agha-mohammadi²

¹ the University of Tokyo, 5-1-5 Kashiwa no ha, Kashiwa, Chiba, Japan,
emmei.tomoki14@ae.k.u-tokyo.ac.jp,

WWW home page: <https://sites.google.com/edu.k.u-tokyo.ac.jp/hflab>

² NASA Jet Propulsion Laboratory, California institute of technology, 4800 Oak Grove Dr,
Pasadena, CA 91109

Abstract. Autonomous exploration of unknown environments with UAVs remains a challenging problem, especially in perceptually degraded environments. Dust, smoke, fog, and a lack of visual or lidar-based features results in severe difficulties for state estimation, navigation, and planning. In this work we consider the problem of autonomous exploration with limited sensors and actuators, via contact-based exploration. We present an accurate force estimation and exploration method for hybrid aerial-ground vehicle. First, we perform modeling and system identification for a hybrid ground and aerial vehicle design which can withstand collisions. Next, better estimation performance using encoder data and IMU measurements is achieved compared to conventional methods and an analysis to choose a appropriate resolution for sensors is presented. Finally, we implement a bouncing control law which encourages exploration down long and narrow passages. We validate our framework in both simulation and hardware experiments.

Keywords: rollocopter, MAVs, force estimation, encoder resolution, hybrid

1 Introduction

1.1 Motivation

As hardware capabilities of MAVs are being developed, their usage is become more widespread in a variety of applications, from rescue to entertainment. Exploration tasks are one main use case of flying robots. For example, the exploration of dangerous mines, caves, urban environments, or wilderness are all highly relevant problems. One particularly interesting use case is the exploration of the planets with atmospheres, from the Mars helicopter mission, to proposed drone exploration missions on Titan, Venus, and more.

Here we build on prior work which was proposed a hybrid aerial-ground vehicle, the "Rollocopter". This hybrid design demonstrates the capability of rolling and flight exploration, with increased energy efficiency and greater capacity for exploration and mobility [1].

In spite of the high adaptability of MAVs, operations in tunnels, caves, or other environments can often be extremely difficult because of degraded conditions such as rough



Fig. 1: Experimental setup: Rollocopter

ground and low visibility. Dark and dusty environments harm visual sensors, which are the most essential tools for autonomous exploration. In particular, dust is a serious problem for the hybrid design because of the tendency of the propellers to kick up dust when near the ground. One good solution for this challenging task is Contact-based navigation.

In this work, we will discuss Contact-based exploration for Rollocopters. We use the Rollocopter design shown on Fig. 1 as the experimental platform, which allows collisions with the environment. As with ordinary MAVs, our Rollocopter also has an IMU on the center of its body. In addition, encoders are installed to each wheel. Hence, collision information can be obtained with them and exploration can be continued by using these sources of information even when all the visual sensors have failed due to dust, fog, smoke, or darkness.

1.2 Literature review - force estimation for MAVs -

So many researches about sensorless force control (e.g. [2, 3]), contact point detection (e.g. [4–6]), and collision reduction control ([7]) have been done for industrial robots.

These days, force related methods are starting to be applied to MAVs. [8] uses the model predictive control for the hybrid system and achieved a exploration with keeping contact along the walls.

[9] provided a external wrench estimation method using IMU and demonstrated a effective control action when the collision occurs. Furthermore, contact point estimation methods are applied to an ordinary MAVs and demonstrated its effectiveness through an experiment.

To solve the task of indoor exploration with MAVs, contact force based exploration methods which allows collision or contact to the environments have been proposed. [10] achieves collaborative transportation task based on a passive force control. Efficiency of "billiards walk exploration" is revealed in [11]. Though a force-sensorless external wrench estimation method for MAVs is proposed in [9], limit for this kind of IMU based force estimation is demonstrated in [12]. Since the IMU is installed at the center of the

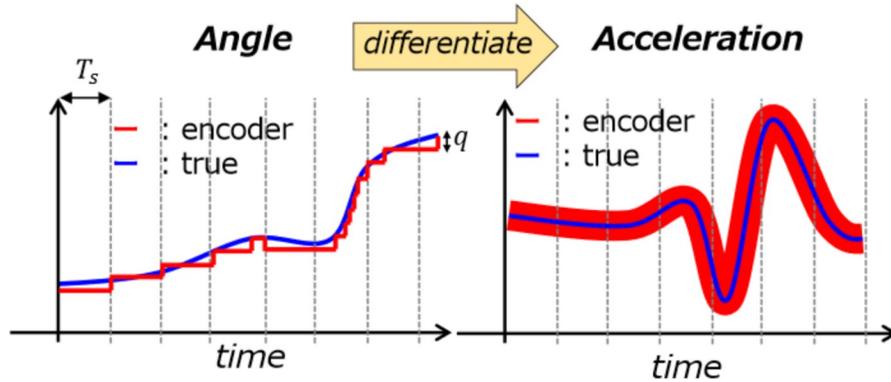


Fig. 2: Problem of quantization noise.

body, IMU based estimation is affected by the perturbation between the IMU and the contact point. [12] revealed that the estimation error becomes quite large with the small force regions such as under 4 N when we use IMU for force estimation. Therefore, they have developed their own ultra light force sensors and attached them at the top of their MAVs.

1.3 Literature review - Sensor detection for force estimation -

Since our Rollocopter have encoder on each wheels, not only odometry information but also wheel force information on the wheels can be obtained. Though high resolution is not required for the odometry detection, its resolution should be discussed carefully when we use encoder in order to detect external torque on the wheels.

To detect external torque on the wheels, wheel encoder signal must be differentiated 2 times to get the acceleration [13]. However, differentiation operation expands the sensor error of the encoder and it causes a large noise called quantization noise on acceleration signal as shown on Fig. 2. This quantization noise should be small enough compared to the signal level so as to use encoder information for the force estimation.

There are many researches about minimal encoder decision for network control (e.g. [14]), quantization noise analysis, and its suppression methods [15, 16]. Those researches are mainly focused on channel capacity constraint or suppressing quantization noise itself.

1.4 Literature review - Contact Inertial Odometry

The legged locomotion research community has used contacts for a long time within kalman filters [17–22] By assuming a fixed contact points of the feet and by leveraging the forward kinematics of the legged and using the joint configurations, it is possible to have measurements. This however assumes no slip, feet usually use additional sensors to sense the ground and detect collisions, and since the robot is walking on the ground, gravity passively helps to maintain the contact, as if it was passively doing force control

to maintain contact. In this work, this is more challenging, as the drone does collisions almost instantly and the information is much more sparse, we have directly slip and the motions can be very fast.

Contacts are also used in manipulation - robotic arms algorithms, where the shape of a body can be reconstructed by simply touching it. Our work is also related to this work [23–28].

A lot of work deals with improving the performance of IMU-only state estimation for drones, such as [29]. Our work focuses on the measurement update coming from the information of a collision, and does not try to achieve state of the art in this field. This work could be used to improve the performance of our kalman filter however.

zero-velocity pseudo-measurements. Here, we do parallel velocity updates, since we can't assume that the drone stays in contact with the wall without moving... EKF Paper for walking with smartphones, very relevant. However, easier since again, this is walking so contact remains. Also these works often assumes the availability of a GPS system, which helps detect when the person stops walking so they can do a pseudo measurements [30–33]

Pseudo measurements for cars are also used, where the no slip holonomic constraint can be leveraged [34]. Our work is related to this work, but we our update is different and can be made in any direction, as it is perpendicular to the wall.

On the planning and control side, many people worked on how to recover from a collision with a drone [35, 36]. However, these works usually try to avoid the collision, whereas we are taking the collision to our advantage. Also, they usually assume basic properties of the wall so they can take it into account..

1.5 Contributions: Keep navigating when everything fails !

In this paper, a novel framework for autonomous guidance, navigation and control of autonomous hybrid vehicles in GPS denied, perceptually-degraded environments is proposed. In fact, we present a novel state estimation method: Collision Inertial Odometry (CIO), which only requires the use of an Inertial Measurement Unit (IMU) and leverages information from contacts as a measurement update in a Kalman Filtering framework. We demonstrate autonomous navigation using only proprioceptive sensors, i.e. an IMU.

First, we leverage existing work in force estimation and extend it for our novel hybrid vehicle platform: Rollocopter. By precisely describing the dynamics of our system, we achieve reliable collision detection, accurate force estimation and precise contact position estimation. Our method is generic, computationally tractable and can easily be implemented on other platforms. We also extend it for the rolling phase of our vehicles and show that it improves the estimation by a lot. It works great and leverages the properties of our system and is novel. Extensive hardware experiments demonstrate the accuracy and reliability of our method, showing that this method can be used always without problems.

Second, we develop a novel state estimation method: Collision Inertial Odometry (CIO), which only requires the use of an Inertial Measurement Unit (IMU) and leverages information from contacts as a measurement update in a Kalman Filtering framework. Using this framework, we bring three research communities together: state

stimulation from legged systems, and aerial drone community which does forces/contact estimation, and we show that we can use these works for very dynamic motions within an EKF framework. We show that it significantly improves the accuracy of the state estimates of the system velocity, such that it can be used to recover control from collision.

Finally, we demonstrate our method in an (bad - TL) environment and show that we can autonomously navigate in a maze. We present a bouncing controller leveraging the advantages of our hybrid vehicles, reducing the drift of the velocity in all directions so that we can traverse rough terrain where a rolling-only navigation would not work, and we can do it at fast speeds since we are flying. This shows that we do not need the assumption of zero velocity, commonly found in similar work, since we only update the perpendicular velocity. Our framework is well suited to dynamic navigation and control of drones.

Finally, we present extension opportunities, since this work paves the way for extensions. This work is just the start of a long road !!!! This work applies to any kind of vehicles, including rolling, flying and hybrid vehicles. In particular, it is very well suited for small computationally-limited quadrotor platforms with low cost cameras, where the estimates from an IMU can significantly improve the accuracy of these algorithms. Also, since our method works with an IMU only, we can handle VIO failures as we demonstrate in hardware experiments

In this paper, a Contact-based exploration method for the hybrid Rollocopters is presented and 3 main contributions are addressed here.

First, accurate force sensorless collision detection method is proposed for Rollocopter by using IMU and encoder. Since our Rollocopter consists 3 parts, body and 2 wheels, physical model becomes more complex compared to the conventional researches and the different modeling method should be used. Especially, not only the IMU but also encoder information is indispensable to estimate the external wrench on Rollocopter. Moreover, this estimation method is more sensitive to the external force than only IMU case. It is because the encoder position is closer to the contact point than IMU and it does not include the perturbation between the body and the contact point.

Secondly, method to determine the required specifications of encoder is demonstrated. Though we've found out that encoder is necessary for the force estimation of Rollocopter, its resolution must be determined carefully as there's a problem of quantization noise. In this paper quantization noise for the force estimation is theoretically formulated according to its encoder resolution and estimation method. Encoder resolution determination method for the wrench estimation is proposed based on them.

Thirdly, Contact-based exploration method for Rollocopter is proposed both for rolling and flying mode considering multi-body modeling and nonholonomic constraint. The force based navigation for MAV is proposed in [12] and exploration with minimal equipment is achieved. Its quite useful in dark and dusty situation where visual sensors do not work. However, for the Rollocopter, force based navigation for the rolling mode should be developed because the flying mode is quite energy consuming. Force control for electric vehicle without force sensor is proposed in [37]. However, it doesn't assume the application for the exploration and doesn't expect collisions. Moreover, in rolling mode, non-holonomic constraint on the wheels should be considered since it can be modeled as differential drive vehicle[38, 39]. In this paper, appropriate non-holonomic

Table 1: Specification of experimental setup

Variable	Parameter	Value
Total mass	m_t	4.036 kg
Wheel mass	m_w	0.283 kg
Inertia	I_x, I_y, I_z	0.09, 0.074, 0.09 kg·m ²
Propeller diameter	D	0.2286 m
Thrust coefficient	C_p	0.11
Torque coefficient	C_q	0.008
Air density	ρ	1.18
Wheel radius	r	0.2667 m
Wheel inertia	J	0.00975 kg·m ²
Arm length	l	0.254 m
Half length of the shaft	L	0.3125 m
Force estimator gain	K_f	10
Torque estimator gain	K_M	10
Wheel torque estimator gain	K_{we}	10
total, body, wheel	Subscript t, b, w	-
right, left	Subscript r, l	-
external, input	Subscript e, in	-

modeling and control strategy is chosen for the Contact-based exploration for the rolling mode.

This paper is organized as follows. In Section 2, we present our accurate modeling of our system (Rollocopter/ hybrid vehicle) and the derived contact force detection and estimation method. This method is tailored to both the flying and rolling modes of our hybrid vehicles (Rollocopter), and thus can be used by both research communities. Leveraging this reliable force estimation method tailored to our vehicle, In Section 3, we present CIO: our collision-aware IMU-only state estimation algorithm. We also show how to do reactive planning and present our control strategy for the hybrid vehicle. Section 4 presents results of each section of this work. We show the reliability and accuracy of our force estimation and detection method and contact point estimation. We also demonstrate the effectiveness of our state estimation odometry algorithm, outperforming IMU-only EKFs. We also demonstrate autonomous navigation in a dark environment, using flying only, bouncing control and rolling. Our conclusion in Section 5 summarizes this paper and provides new promising directions of research for different research communities, since our work opens new areas of research.

We also present novel results for sensor selection based on fundamental theoretical results, presented in the Appendix A.

2 System Modeling and Contact Forces Estimation

2.1 Experimental setup

Table 1 shows the specifications of the Rollocopter used in this paper. This Rollocopter is kind of the coaxial octorotor, which has two large wheels for rolling on the ground. Rollocopter can choose rolling mode and flight mode depending on an environment.

2.2 Thrust model

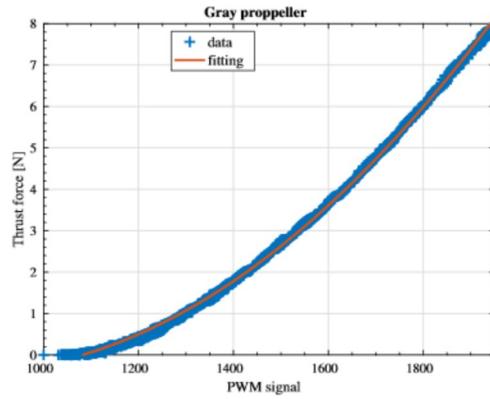


Fig. 3: Modeling of propeller thrust force.

Aerodynamic wrench on each propeller can be formulated as:

$$T_i = \rho C_p D^4 \bar{n}_i^2 = C_T \bar{n}_i^2 \quad (1a)$$

$$Q_i = \rho C_q D^5 \bar{n}_i^2 = C_Q \bar{n}_i^2 \quad (1b)$$

Here, each \bar{n}_i , T_i , and Q_i describes the rotation speed, thrust, and rotation torque of i th propeller. Squared propeller speed vector $(\bar{n}_1^2, \bar{n}_2^2, \bar{n}_3^2, \bar{n}_4^2, \bar{n}_5^2, \bar{n}_6^2, \bar{n}_7^2, \bar{n}_8^2)^T$, can be transformed to input wrench $(f_{inz}, M_{inx}, M_{iny}, M_{inz})^T$ with the following transformation matrix \mathbf{C} .

$$\mathbf{C} = \begin{pmatrix} C_T & C_T \\ -1C_T & 1C_T & 1C_T & -1C_T & 1C_T & -1C_T & -1C_T & 1C_T \\ -1C_T & -1C_T & 1C_T & 1C_T & -1C_T & -1C_T & 1C_T & 1C_T \\ -C_Q & C_Q & -C_Q & C_Q & C_Q & -C_Q & C_Q & -C_Q \end{pmatrix} \quad (2)$$

The setup uses a brushless DC-motors are used and it has a nonlinear relationship between their PWM input and their rotation speed [40]. Therefore, we conducted an experiment to measure the propeller characteristic and transform PWM signals into thrust force directly based on quadratic approximation. Fig. 3 shows the measurement result.

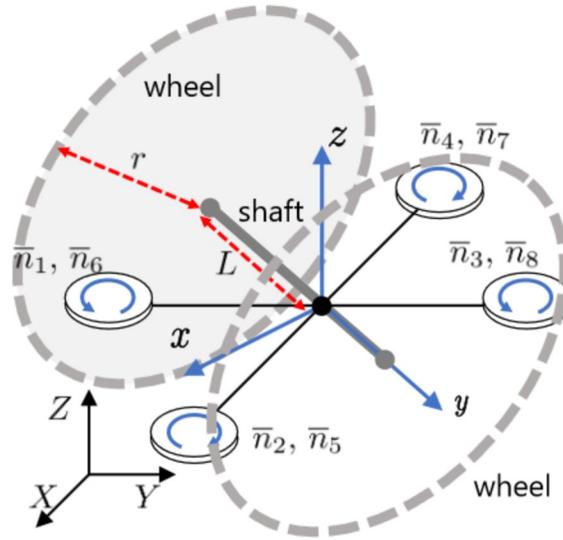


Fig. 4: Geographical model of the quadcopter

2.3 Multi-body modeling

This Rollocopter is a multi-rigid body system which has the 3 parts: the body and 2 wheels. The physical model of the Rollocopter is formulated based on Kane's motion equation. In flying mode, the experimental model is formulated as:

$$\mathbf{m}_i(\dot{\mathbf{v}} + \boldsymbol{\gamma} \times \mathbf{v}) = \mathbf{f}_{in} + \mathbf{f}_e \quad (3a)$$

$$\mathbf{I}\dot{\boldsymbol{\gamma}} + \mathbf{I}_0\dot{\boldsymbol{\gamma}} + \begin{pmatrix} 2m_w L^2(\dot{\gamma}_x + \gamma_z \gamma_y) \\ J(\dot{\omega}_l + \dot{\omega}_r) \\ 2m_w L^2(\dot{\gamma}_z - \gamma_x \gamma_y) \end{pmatrix} + \boldsymbol{\gamma} \times \mathbf{I} \cdot \boldsymbol{\gamma} = \mathbf{M}_{in} + \mathbf{M}_e \quad (3b)$$

$$J(\dot{\gamma}_y + \dot{\omega}_i) = M_{wei} \quad (3c)$$

(3a) is translational, (3b) is rotational, and (3c) is wheel rotation motion equation[41]. Here, the frame is fixed on the body and \mathbf{v} , $\boldsymbol{\gamma}$, $\boldsymbol{\omega}$, \mathbf{f} , and \mathbf{M} respectively means the velocity, the body angular velocity, the wheel rotation speed, and the force. Subscript *in* means "input thrust force", *e* means "external", *w* means "wheel", and *i* = (*l*, *r*) means either "left" or "right". $\dot{\mathbf{v}} + \boldsymbol{\gamma} \times \mathbf{v}$ can be directly measured by the IMU. The translational acceleration or the body angular velocity can be obtained with IMU and the wheel angular velocity can be obtained with encoder on each wheels.

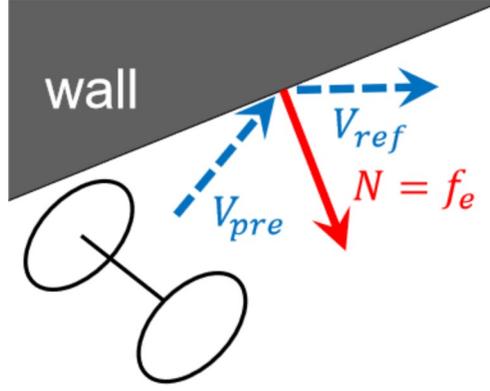


Fig. 5: Control strategy of force based exploration.

In (3b) and (3c), M_e and M_{wei} can be calculated as:

$$\mathbf{M}_e = \begin{pmatrix} L(f_{wezl} - f_{wezr}) - \sum_{i=l,r} r_{ezi}f_{weyi} \\ \sum_{i=l,r} (r_{ezi}f_{wexi} - r_{exi}f_{wezi}) \\ L(f_{wexr} - f_{wexl}) + \sum_{i=l,r} r_{exi}f_{weyi} \end{pmatrix} \quad (4a)$$

$$M_{wei} = r_{ezi}f_{wexi} - r_{exi}f_{wezi} \quad (4b)$$

In (4a)-(4b), we assume all the collisions occur on wheels.

In the rolling mode, pitch motion of the body affects the estimation results which is different from the flying mode. For example, even when the wheel is fixed on the ground, the body still have the degree of freedom in pitch motion. Therefore, Rolling frame whose z axis is aligned with the ground should be introduced. Rolling frame is fixed on the center of gravity, its x' axis faces the same direction as the proceeding direction, y' axis is same as the body frame, and z' axis is aligned with the ground.

2.4 Force estimation and collision detection

Force estimation in flying mode In order to achieve accurate estimation, both IMU and encoder information should be combined. Based on (3a)-(3c), external force, torque, and moment on the wheels can be calculated as:

$$\hat{\mathbf{f}}_e = K_f \int_0^t (\mathbf{m}_t \mathbf{a} - \mathbf{f}_{in} - \hat{\mathbf{f}}_e) \quad (5a)$$

$$\hat{\mathbf{M}}_e = K_M \left(\mathbf{I} \boldsymbol{\gamma} + \begin{pmatrix} 2mL^2(\dot{\gamma}_x + \gamma_z \gamma_y) \\ J(\dot{\omega}_l + \dot{\omega}_r) \\ 2mL^2(\dot{\gamma}_z - \gamma_x \gamma_y) \end{pmatrix} + \int_0^t (\boldsymbol{\gamma} \times \mathbf{I} \cdot \boldsymbol{\gamma} - \mathbf{M}_{in} - \hat{\mathbf{M}}_e) dt \right) \quad (5b)$$

$$\hat{M}_{wei} = K_w \left(J(\gamma_y + \omega_y) - \int_0^t \hat{M}_{wei} dt \right) \quad (5c)$$

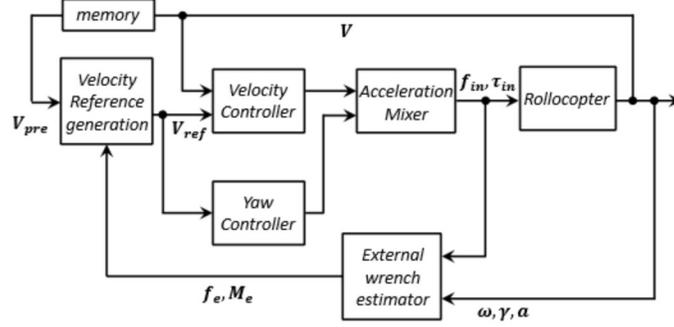


Fig. 6: Block diagrams of the bouncing control

Force estimation in rolling mode At the rolling mode, following non-holonomic constraint is applied:

$$\dot{v}_x = \frac{r}{2}(\omega_r + \omega_l) \quad (6a)$$

$$\dot{v}_y = 0 \quad (6b)$$

$$\gamma_z = \frac{R}{2L}(\omega_r - \omega_l) \quad (6c)$$

In addition, though body frame, which is fixed on the Rollocopter body is adopted in flying mode, rolling frame, whose z axis is aligned with the ground should be introduced here because it doesn't have vertical and roll motion. From those constraints and (3a), we can derive following estimation formula for rolling mode:

$$\hat{f}_{ex} = \frac{rm_t}{4} K_{fx} \int_0^t ((\dot{\omega}_r + \dot{\omega}_l) - f_{inx} - f_{ex}) dt \quad (7a)$$

$$\hat{f}_{ey} = \frac{r^2 m_l}{4L} (K_{fy} \int_0^t (\omega_r^2 - \omega_l^2 - \hat{f}_{ey})) \quad (7b)$$

$$\hat{M}_{ez} = \frac{r(I_z + I_{0z} + 2m_w L^2)}{2L} K_{Mz} \int_0^t ((\dot{\omega}_r - \dot{\omega}_l) - M_{inz} - \hat{M}_{ez}) \quad (7c)$$

Here, \mathbf{a} means the translational acceleration obtained by IMU.

Collision detection For the collision detection, only the external translational force information is required if the estimation accuracy is enough. Most of the case, we can easily threshold the signal from the noise because the collision force is much larger than the noise level. However, the force estimation in (5a) only includes the IMU information. Perturbation between the contact point and the IMU deteriorates the quality of the estimation as addressed in [12] and some kind of collisions cannot be detected when only the IMU value is used.

One way to reflect encoder information on the collision detection is to introduce following evaluation function $W[k]$:

$$W[k] = w_f \|\hat{\mathbf{f}}_e[k]\|^2 + w_M \|\hat{\mathbf{M}}_e[k]\|^2 + w_{M_{we}} (\hat{M}_{wl}[k]^2 + \hat{M}_{wr}[k]^2) \quad (8)$$

Here, $w_f = 1$, $w_M = 1/L^2$, $w_{tau_{we}} = 1/r^2$ is chosen so as to make $W[k]$ force dimension for example. When this value goes over a threshold, we define it collision.

Contact point detection Contact point detection on rolling mode is quite useful when we want to separate the collision force from the rough ground. Based on the estimated external wrench, contact point on each wheels can be detected. Here, we introduce 2 assumptions: the contact force on each wheels \mathbf{f}_{wi} occurs only on the edge of the wheels and the lateral collision occurs only on the one side of wheel. It is because non-holonomic constraint of $V_{y'} = 0$ holds on. In addition, contact point to the ground always exists at the rolling mode. Therefore,

$$\mathbf{f}_{we} = \mathbf{f}_{wel} + \mathbf{f}_{wer}, \quad r_{xi}^2 + r_{zi}^2 = r^2 \quad (9)$$

holds. By solving simultaneous quadratic equation the relationship (4a)-(4b), (5a)-(5c), and (9), contact points \mathbf{r}_i can be calculated.

2.5 Encoder resolution determination

Since the external torque can be calculated by the product of the wheel acceleration and the wheel inertia, we should know the noise level of 2 times pseudo differential and determine the appropriate encoder resolution. This can be determined as:

$$R \geq \log_2 \left[\frac{2\pi J}{\tau^2 S} \sqrt{\frac{3e^{\frac{T_s}{\tau}}}{4 + e^{-\frac{T_s}{\tau}}}} \right] \quad (10)$$

The details of this equation is based on (27) in appendix . Here, R and S respectively shows the required encoder resolution [bit] and signal level [N]. We can obtain the signal level of the force from the force estimation experiment. Here, for example, we put the detection threshold around 0.05 Nm and assume that the collision finishes during 100 ms. The threshold for detecting collision should be larger than 3σ noise level and the cutoff frequency of the pseudo differential should be faster than 100 ms. By substituting T_s with a sampling period of 10 ms, τ with 100 ms, and $3 * S$ with $0.05/J$ in (27), we obtain the appropriate quantization width of 0.0074. This applies 7.1865 bit.

3 Contact Inertial Odometry, Control and Planning

3.1 Contact Inertial Odometry

As discussed in the introduction, it is possible to take information about contacts to update the state estimation as a measurement update within a Kalman Filtering framework. This is what we present in this section.

We use the Robot Localization ROS package [42] as a baseline for our algorithm. Given a robot pose described by its state \mathbf{x}_k at time k , with its discrete time dynamics corrupted by Gaussian-distributed noise

$$\mathbf{x}_{k+1} = f(\mathbf{x}_k) + \mathbf{w}_k$$

, where $\mathbf{w}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}_k)$, with $\mathbf{Q} \succ 0$ the process noise covariance, it implements an Extended Kalman Filter for which the prediction step is given as

$$\hat{\mathbf{x}}_{k+1} = f(\hat{\mathbf{x}}_k) \quad (11a)$$

$$\hat{\mathbf{P}}_{k+1} = \mathbf{F}_k \hat{\mathbf{P}}_k \mathbf{F}_k^T + \mathbf{Q}_k, \quad (11b)$$

where \mathbf{F}_k denotes the Jacobian matrix of $f(\cdot)$.

Note that it does not estimate biases of sensors, e.g., of the Inertial Measurement Unit (IMU).

Given a measurement of the state

$$\mathbf{z}_k = h(\mathbf{x}_k) + \mathbf{v}_k, \quad (12)$$

where $\mathbf{v}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{R}_k)$, with $\mathbf{R} \succ 0$ the measurement noise covariance, it is possible to perform an update to the EKF as

$$\mathbf{K}_k = \hat{\mathbf{P}}_k \mathbf{H}_k^T (\mathbf{H}_k \hat{\mathbf{P}}_k \mathbf{H}_k^T + \mathbf{R}_k)^{-1} \quad (13a)$$

$$\mathbf{x}_{k+1} = \hat{\mathbf{x}}_{k+1} + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}_k) \quad (13b)$$

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k) \hat{\mathbf{P}}_k (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)^T + \mathbf{K}_k \mathbf{R}_k \mathbf{K}_k^T, \quad (13c)$$

where \mathbf{H}_k denotes the Jacobian matrix of $h(\cdot)$

After a collision, we assume the velocity of the robot to be parallel to the collided obstacle. Given a previous velocity \mathbf{v}_{pre} , the parallel velocity \mathbf{v}_{\parallel} is therefore computed as

$$\mathbf{v}_{\parallel} = \mathbf{v}_{pre} - \frac{(\mathbf{v}_{pre} \cdot \mathbf{N})}{\mathbf{N} \cdot \mathbf{N}} \mathbf{N}. \quad (14)$$

Therefore, we introduce the pseudo-measurement model for parallel collision velocity updates as

$$\mathbf{z} = \mathbf{v} + \boldsymbol{\varepsilon} \leftarrow \mathbf{v}_{\parallel}, \quad (15)$$

with $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma}_{\boldsymbol{\varepsilon}})$ Gaussian-distributed iid noise and \mathbf{v}_{\parallel} defined in Equation (14).

Note that as collisions involve a loss of energy. Therefore, this model is not biased (mention something here).

Assuming known friction coefficients between the wall and the robot, it may be possible to estimate this loss of energy in the parallel direction of the contact (as if there was drag.). Since we operate in unknown complicated challenging environments, this is not investigated in this paper and can be explored in future work.

Despite the simple looks of our approach, this method showed to provide very good results, as shown in the results section. They were so good that we managed to get a planner to work at the same time, which we present next.

3.2 Reactive Planning

Using the detected forces, we present a simple planner to illustrate our approach and reach the end of a tunnel or a maze in spite of dust in the environment. This work is motivated by billiard paper work, which proved that the end of a maze can be reached with probability one by trying random actions. Hence, a simple planner is good enough, and this will be verified in the results section.

The first planning method investigated is the following

$$\mathbf{v}_{ref} = \mathbf{v}_{pre} - 2 \frac{(\mathbf{v}_{pre} \cdot \mathbf{N})}{\mathbf{N} \cdot \mathbf{N}} \mathbf{N} \quad (16)$$

However, this method relies on accurate knowledge of the previous velocity \mathbf{v}_{pre} , which may be inaccurate. Furthermore, it is not clear where to go when arriving against the wall with a perpendicular velocity.

Since the previous reference velocity and collision forces are both expressed in body frame, it is better to directly compute everything in body frame, with the unique goal of going forward while slightly avoiding obstacles. Therefore, we define the next reference velocity as the projection of the collision force onto a cone in the direction that we want to go to

$$\psi = \arctan(\mathbf{v}_{prev}, \mathbf{N}) \quad (17a)$$

$$\mathbf{v}_{next} = \mathbf{R}(\psi) \mathbf{N} \quad (17b)$$

Finally, velocities may drift in vertical directions. Therefore, it may be better to periodically go down to touch the ground and update those velocities. We do so every 2 seconds and also bounce against the ground and the ceiling if we touch it. The planner output is therefore given as in Eq. (17).

Here, \mathbf{N} means a normal force from the wall, which can be estimated by the proposed method. \mathbf{v}_{pre} and \mathbf{v}_{ref} respectively means a pre-collision speed and desired reference velocity.

For the rolling mode, since ground vehicles are bound by non-holonomic constraints, the Rollocopter cannot move toward yaxis as the flying mode and then (16) cannot be applied directly. Therefore, when the collision is detected, the Rollocopter (i) goes back in order to make enough space for turning, (ii) turns the direction based on (16), and (iii) goes straight toward the direction of \mathbf{V}_{ref} .

In this bouncing action, the Rollocopter just goes back when the front collision occurs. However, it is not efficient for the exploration. Therefore, it turns 90 deg and goes along the wall when the front collision is detected.

3.3 Low-Level Control

Fig. 6 shows the block diagram of the low layer controller. Velocity reference can be generated as (16) so that Rollocopter would bounce like a billiards[11].

Table 2: Ground truth of the collision time (flying mode).

time	20s	24s	30s	34s	53s	61s	67s	78s	85s	90s
part (angle)	F (0°)	F (0°)	FW (0°)	FU (0°)	FL (45°)					

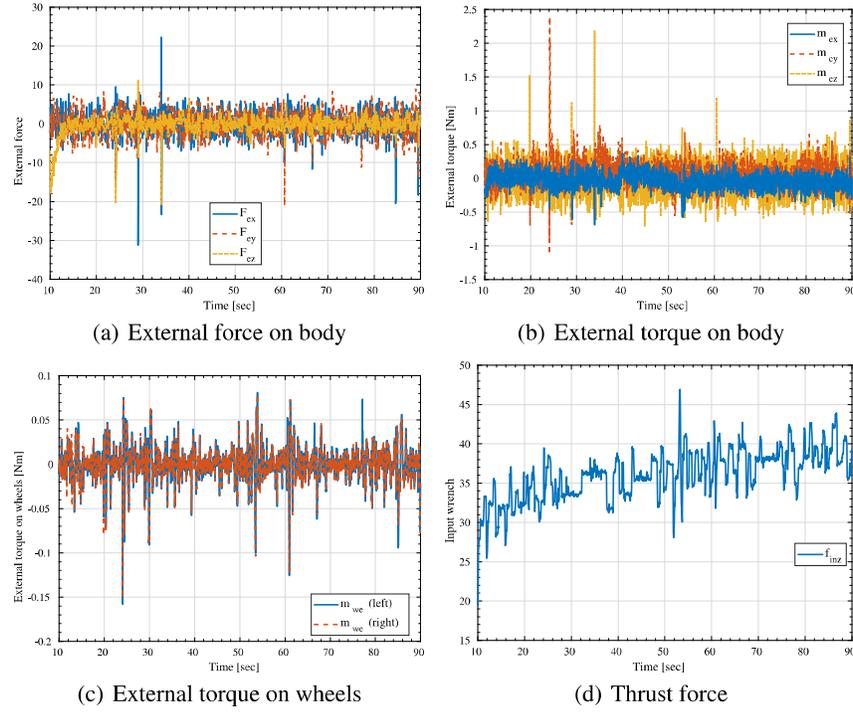


Fig. 7: external force estimation (flying).

4 Results

4.1 External wrench estimation for Rollocopter

A force estimation experiment was conducted to evaluate the estimation method. The Rollocopter was operated manually and hit against the wall with various attitude and direction during the experiment. Here, high-pass filter is applied to remove the offset of IMU.

Fig. 7 shows the experimental result of force estimation and Table 2 shows the ground truth of the collision time. "F", "W", "U", and "L" respectively means part of collision "front", "lower part", "upper part", and "left side" and bracketed contents mean the yaw direction of the collision. In Fig. 7(a) and Fig. 7(b), almost all spikes correspond with the time of the collision, which means collision detection worked correctly. Moreover, by comparing the amplitude of each axis, force direction can be

Table 3: Ground truth of the collision time (rolling).

time	16s	26s	32s	34s	41s	52s	78s
position (direction)	F (0°)	F (0°)	B (180°)	F (0°)	F (0°)	F (0°)	F (0°)

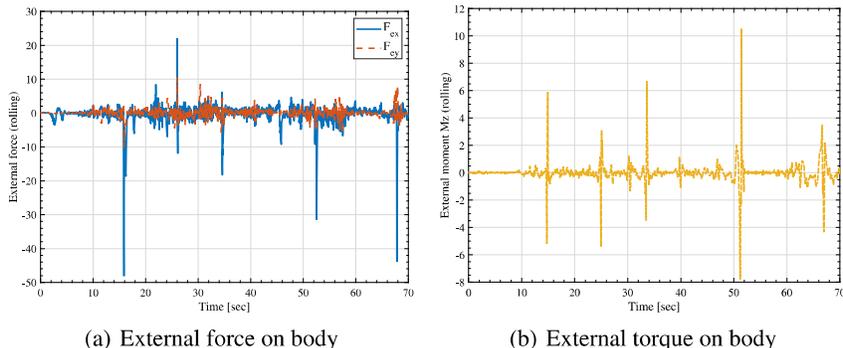


Fig. 8: External force estimation (rolling).

known. For example, collision at 30s mainly includes negative value of f_x and f_z because its front-upper part hit the wall.

However, at 53s, IMU based force estimation failed to detect the collision force here. Although there is a small spike at 53s in Fig. 7(b), it is too small to distinguish from the noise and the threshold for the collision detection is too severe. On the other hand, estimated wheel torque succeeds in detecting this small forces whereas both body force and torque estimation failed to detect the collision at 53s. The amplitude of the spike is large enough to tell apart from the noise. Wheel torque estimation is good at detecting small and low frequency forces, which is difficult for the IMU based estimation.

Another experiment for rolling mode is conducted. Fig. 8 shows the experimental result and the ground truth of the collision time is on Table 3. Timing and the direction of the estimated force is appropriate even in the rolling mode.

4.2 Contact point detection

Experiment for contact point detection is also conducted with rolling mode. Both wheels of Rollocopter is collided from the front with the box whose height is 15 cm with the rolling mode.

Fig. 9 shows the experimental result. Since the wheel radius is 26.7 cm, value of r_z should be around 0.117 m ($= 26.7 - 15$ cm). However, measured value is smaller than the actual height. This is because the estimated contact point is based on the combined force of the collision force and the normal force from the ground. Therefore, point of contact is underestimated than the actual value. However, this result is enough to distinguish whether a contact is caused by collision or by terrain.

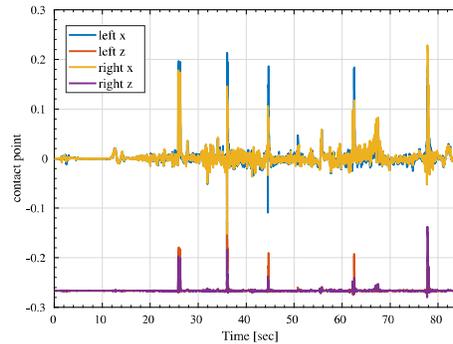


Fig. 9: contact point detection (experiments)

4.3 CIO: Collision Inertial Odometry

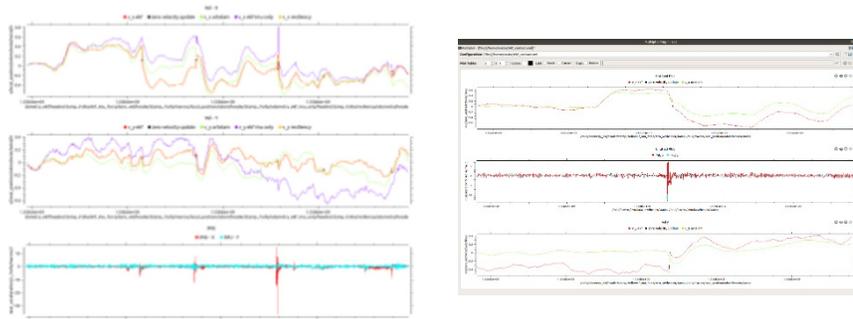


Fig. 10: CIO EKF results for flying mode. Fig. 11: CIO EKF results for ground mode.

4.4 CIO and planning

Going through a maze In this section, we show that it is possible to traverse a cluttered environment using an IMU only. We leverage our contact detection and estimation, CIO, and the reactive planner on our hybrid vehicle platform.

In Figure 13, we show the maze environment.

Note that the quick divergence of the EKF using the propagation of an IMU only, without CIO. This indicates that the drone would have crashed without our algorithm, demonstrating the effectiveness of our approach.

Bouncing Control Since the velocity can drift in vertical directions, it is beneficial to impose periodical touchdowns to obtain measurements of the vertical speed. This

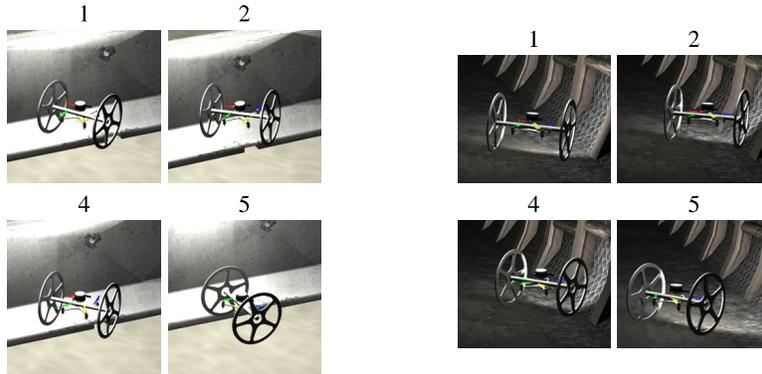


Fig. 12: Experimental result of the Contact-based navigation for Rollocopter



(a) Left part of the maze, including start position. (b) Right part of the maze, including end position.

Fig. 13: Maze Environment

shows the advantages of *perception-aware planning*, where the planning module is designed to improve state estimation properties.

5 Conclusion

5.1 Summary

In this paper, we have realized the force based navigation for ground-aerial hybrid vehicles, Rollocopter. Not only IMU but also wheel encoders are used to estimate the external wrench on the Rollocopter and it worked both rolling and flight mode. Moreover, appropriate encoder resolution is determined for the force estimation. By combining these methods, Contact-based navigation for Rollocopter is achieved at both rolling and flying mode. In this work, blind navigation for Rollocopter is demonstrated with force sensor-less approach. Future work includes sensor-less contact force control based on estimation method proposed in this paper. Not only encoder specification design but also IMU or other sensors should be also addressed.

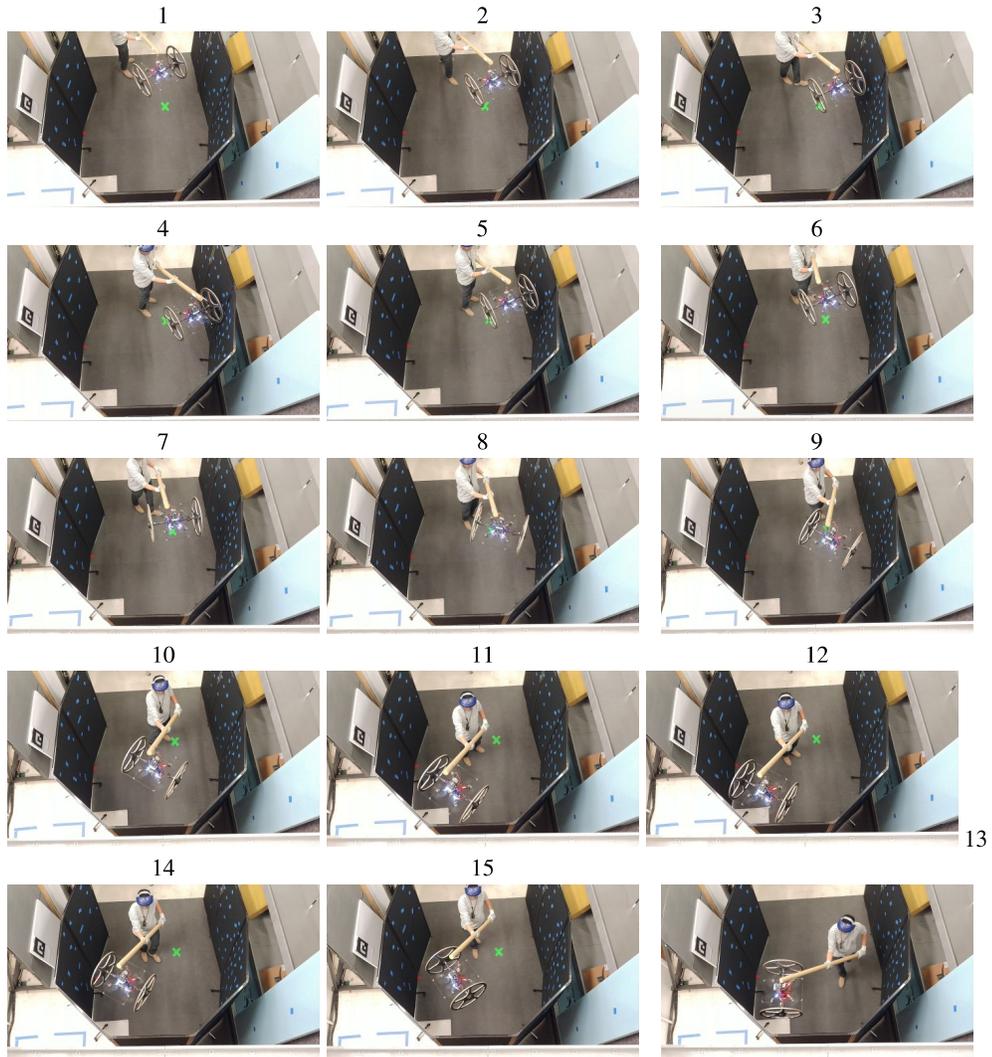


Fig. 15: Reactive Planning and Control using CIO and our reactive planner. We show that it can autonomously bounce off walls and react to external forces, without visual odometry.

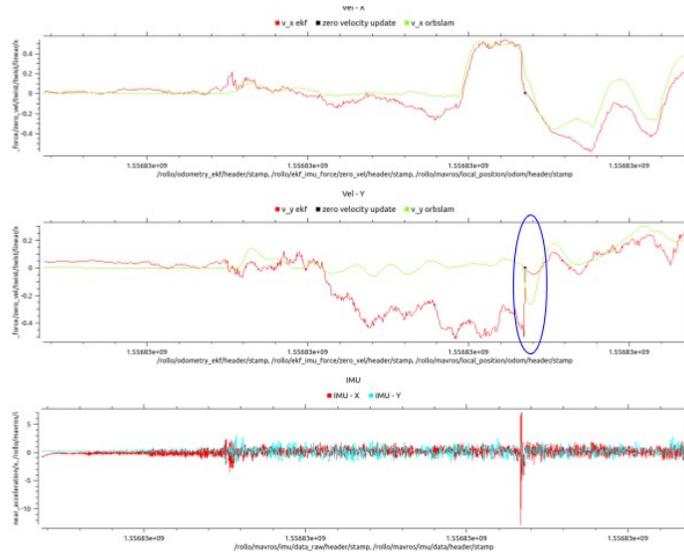


Fig. 14: Parallel velocity measurement updates. We show that we can accurately correct the state estimation error after a collision.

5.2 Future work

In future work, we will exploit perception-aware planning to ensure that the drift is not too high and that the state estimates are accurate enough to stabilize and control the drone. We will also compute theoretical time bounds for the state error to diverge, such that we know exactly how long we can fly before requiring a collision or a landing.

Also, exciting new areas of research including active collision-based mapping. Using a prior map of the environment, it would be possible to localize the autonomous agent in positions (instead of only in velocity as in this work), so a robot can traverse a maze using the prior map and this novel CIO. This would enable autonomous robots such as Shapeshifter to be able to return to its original starting point in spite of loss of visual odometry, assuming a prior map is available, which is the case since it was able to get to this point in space.

Finally, improving the state estimator is possible and we will investigate this as well. First, exploiting rigid body kinematics as in legged locomotion is possible. For instance, after a collision, it is possible to exploit the model of the drone (for instance with rigid wheels) to infer the resulting velocity, assuming a pivot point at the collision point for instance. Also, collisions with soft surfaces, accounting for loss of energy, friction against walls are also possible to be taken into account to improve the estimator. Finally, our EKF is very basic and doesn't include biases. It would be interesting to investigate error-state kalman filters, for instance the ones using manifold-type constraints, to improve the accuracy of the estimator. Also, the covariance from the perceived estimation accuracy of the contact should be tuned, using machine learning methods for instance.

Acknowledgement

This research was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

T. Lew is partially supported by the 242 Program of the Hubert Tuor Foundation.

References

1. D.S. Elliott, A. Tagliabue, pp. 1–8 (2019)
2. T. Yoshioka, T.T. Phuong, K. Ohishi, 2015 IEEE International Conference on Mechatronics (ICM) (1), 100. DOI 10.1109/ICMECH.2015.7083955
3. S. Oh, K. Kong, IEEE/ASME TRANSACTIONS ON MECHATRONICS **22**(1), 71 (2017)
4. R. Featherstone, S. Sonck Thiebaut, O. Khatib, in *Proceedings of the 1999 IEEE International Conference on Robotics and Automation* (1999), pp. 3281–3286
5. N. Kurita, S. Sakaino, T. Tsuji, IEEE Transactions on Industry Applications **134**(5), 517 (2014). DOI 10.1541/ieejias.134.517
6. K. Kutsuzawa, S. Sakaino, T. Tsuji, in *IECON Proceedings (Industrial Electronics Conference)* (2016). DOI 10.1109/IECON.2016.7793932
7. E. Sariyildiz, K. Ohnishi, IEEE/ASME Transactions on Mechatronics **20**(2), 750 (2015). DOI 10.1109/TMECH.2014.2321014
8. E. Gryazina, B. Polyak, European Journal of Operational Research (2014). DOI 10.1016/j.ejor.2014.03.041
9. T. Tomić, C. Ott, S. Haddadin, IEEE Transactions on Robotics (2017). DOI 10.1109/TRO.2017.2750703
10. A. Tagliabue, M. Kamel, S. Verling, R. Siegart, J. Nieto, in *Proceedings - IEEE International Conference on Robotics and Automation* (2017). DOI 10.1109/ICRA.2017.7989678
11. K. Alexis, G. Darivianakis, M. Burri, R. Siegart, Autonomous Robots (2016). DOI 10.1007/s10514-015-9485-5
12. A. Briod, P. Kornatowski, A. Klapotocz, A. Garnier, M. Pagnamenta, J.C. Zufferey, D. Floreano, IEEE International Conference on Intelligent Robots and Systems pp. 3987–3992 (2013). DOI 10.1109/IROS.2013.6696926
13. M. Yoshimura, H. Fujimoto, IEEE Transactions on Industry Applications **181**(3), 721 (2012). DOI 10.1541/ieejias.131.721
14. Y. Saigo, H. Okajima, N. Matsunaga, in *2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)* (2018), pp. 1423–1428. DOI 10.23919/SICE.2018.8492662
15. H. Zhu, H. Fujimoto, IEEE Transactions on Industrial Informatics **10**(2), 1361 (2014). DOI 10.1109/TII.2014.2307195
16. A. Suzumura, Y. Fujimoto, T. Murakami, R. Oboe, IEEE Transactions on Industrial Electronics **63**(12), 7607 (2016). DOI 10.1109/TIE.2016.2597133
17. R. Hartley, M.G. Jadidi, J.W. Grizzle, R.M. Eustice, arXiv:1805.10410 [cs] (2018). URL <http://arxiv.org/abs/1805.10410>. ArXiv: 1805.10410
18. M. Benallegue, F. Lamiroux, International Journal of Humanoid Robotics **12**(03), 1550025 (2015). DOI 10.1142/S0219843615500255. URL <http://www.worldscientific.com/doi/abs/10.1142/S0219843615500255>
19. N. Rotella, M. Bloesch, L. Righetti, S. Schaal, in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, Chicago, IL, USA, 2014), pp. 952–958. DOI 10.1109/IROS.2014.6942674. URL <http://ieeexplore.ieee.org/document/6942674/>

20. S. Kuindersma, R. Deits, M. Fallon, A. Valenzuela, H. Dai, F. Permenter, T. Koolen, P. Marion, R. Tedrake, *Autonomous Robots* **40**(3), 429 (2016). DOI 10.1007/s10514-015-9479-3. URL <http://link.springer.com/10.1007/s10514-015-9479-3>
21. M.F. Fallon, M. Antone, N. Roy, S. Teller, in *2014 IEEE-RAS International Conference on Humanoid Robots* (IEEE, Madrid, Spain, 2014), pp. 112–119. DOI 10.1109/HUMANOIDS.2014.7041346. URL <http://ieeexplore.ieee.org/document/7041346/>
22. M. Bloesch, M. Hutter, M. Hoepflinger, S. Leutenegger, C. Gehring, C. David Remy, R. Siegwart, (2012). DOI 10.15607/RSS.2012.VIII.003
23. M.C. Koval, N.S. Pollard, S.S. Srinivasa, in *Robotics Research*, vol. 114, ed. by M. Inaba, P. Corke (Springer International Publishing, Cham, 2016), pp. 375–391. DOI 10.1007/978-3-319-28872-7_22. URL http://link.springer.com/10.1007/978-3-319-28872-7_22
24. S. Li, S. Lyu, J. Trinkle, W. Burgard, in *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, Hamburg, Germany, 2015), pp. 5059–5064. DOI 10.1109/IROS.2015.7354089. URL <http://ieeexplore.ieee.org/document/7354089/>
25. A. Petrovskaya, O. Khatib, *IEEE Transactions on Robotics* **27**(3), 569 (2011). DOI 10.1109/TRO.2011.2138450. URL <http://ieeexplore.ieee.org/document/5784199/>
26. K. Gadeyne, H. Bruyninckx, p. 6
27. T. Debus, P. Dupont, R. Howe, p. 10
28. M.C. Koval, M.R. Dogar, N.S. Pollard, S.S. Srinivasa, in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, Tokyo, 2013), pp. 4541–4548. DOI 10.1109/IROS.2013.6697009. URL <http://ieeexplore.ieee.org/document/6697009/>
29. D. Abeywardena, S. Kodagoda, G. Dissanayake, R. Munasinghe, *IEEE Robotics & Automation Magazine* **20**(4), 32 (2013). DOI 10.1109/MRA.2012.2225472. URL <http://arxiv.org/abs/1509.03388>. ArXiv: 1509.03388
30. E. Foxlin, *IEEE Computer Graphics and Applications* **25**(6), 38 (2005). DOI 10.1109/MCG.2005.140. URL <http://ieeexplore.ieee.org/document/1528431/>
31. B. Wagstaff, V. Peretroukhin, J. Kelly, 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN) pp. 1–8 (2017). DOI 10.1109/IPIN.2017.8115947. URL <http://arxiv.org/abs/1707.01152>. ArXiv: 1707.01152
32. B. Wagstaff, J. Kelly, arXiv:1807.05275 [cs] (2018). URL <http://arxiv.org/abs/1807.05275>. ArXiv: 1807.05275
33. A. Solin, S. Cortes, E. Rahtu, J. Kannala, arXiv:1703.00154 [cs, stat] (2017). URL <http://arxiv.org/abs/1703.00154>. ArXiv: 1703.00154
34. M. Brossard, A. Barrau, S. Bonnabel, (2019). URL <http://arxiv.org/abs/1904.06064>
35. G. Dicker, F. Chui, I. Sharf, in *2017 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, Singapore, Singapore, 2017), pp. 5830–5836. DOI 10.1109/ICRA.2017.7989685. URL <http://ieeexplore.ieee.org/document/7989685/>
36. F. Chui, G. Dicker, I. Sharf, in *2016 International Conference on Unmanned Aircraft Systems (ICUAS)* (IEEE, Arlington, VA, USA, 2016), pp. 717–726. DOI 10.1109/ICUAS.2016.7502535. URL <http://ieeexplore.ieee.org/document/7502535/>
37. T. Enmei, H. Fujimoto, Y. Hori, *IECON Proceedings (Industrial Electronics Conference)* pp. 6435–6440 (2016). DOI 10.1109/IECON.2016.7793970
38. S. Adilolu, *Heavy Metal Removal with Phytoremediation i(tourism)*, 13 (2016). DOI <http://dx.doi.org/10.5772/57353>
39. H. Mirzaeinejad, A.M. Shafei, *Robotica* **36**(10), 1551 (2018). DOI 10.1017/S0263574718000565
40. M. Anderson, K.C. Wong, P. Hendrick, in *Australian Aerospace Congress* (2017), pp. 26–28
41. A.C.C. of Japan (ed.), *Handbook for Physics and Control of Satellite (Japanese)* (Baifukan, 2007)
42. T. Moore, D. Stouch, in *Proceedings of the 13th International Conference on Intelligent Autonomous Systems (IAS-13)* (Springer, 2014)

43. M. Tsutomu, K. Ryo, H. Shinji, *Basic digital control (Japanese)* (CORONA Publishing CO., LTD, 1987)

A Sensor decision

Wheel torque estimation includes 2 times pseudo differential and this causes the noise on the estimated wheel torque. In this section, our purpose is to derive the magnitude of the output quantization noise.

A.1 General formulation of noise prediction

Here, general discussion of the quantization noise amplitude $\sigma\{e[k]\}$ caused by input noise ξ is addressed based on discrete Lyapunov equation[43]. $\sigma\{X\}$ means standard deviations of X . (18) is the discrete time state equation driven by white noise $\xi[k]$ at k th time step.

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k] + \mathbf{b}\xi[k], \quad e[k] = \mathbf{C}\mathbf{x}[k] + d\xi[k] \quad (18)$$

$\mathbf{x}[k]$, $e[k]$, \mathbf{A} , \mathbf{B} , \mathbf{C} , and D each describes state vector, output vector, system matrix, input matrix, output matrix, and feed forward matrix. From (18),

$$E\{\mathbf{x}\xi[k]\} = E\{\mathbf{A}^i \mathbf{x}_0 + \sum_{j=0}^{i-1} \mathbf{A}^{i-1-j} \mathbf{b}\xi[j]\} = \mathbf{0}. \quad (19)$$

holds. $E\{X\}$ means the expected value of X . Output noise covariance $E\{e[k]^2\}$ caused by ξ can be derived as:

$$E\{e[k]^2\} = \mathbf{c}E\{(\mathbf{x}[k]\mathbf{x}^T[k])\}\mathbf{c}^T + E\{\xi[k]^2\}d^2. \quad (20)$$

In the steady state, since $\mathbf{X}_{i+1} = \mathbf{X}_i = \mathbf{X}$ holds,

$$\mathbf{X}_{i+1} = \mathbf{A}\mathbf{X}_i\mathbf{A}^T + \mathbf{b}V_\xi\mathbf{b}^T \quad (21)$$

can be derived. Here, $\mathbf{X}_i \equiv E\{(\mathbf{x}[k]\mathbf{x}^T[k])\}$, $V_\xi \equiv E\{\xi[k]^2\}$. With the solution of this (21), output quantization noise

$$\sigma\{e[k]\}^2 = E\{e[k]^2\} = \mathbf{c}\mathbf{X}\mathbf{c}^T + V_\xi d^2 \quad (22)$$

is calculated.

A.2 Quantization noise of wheel torque estimation

Former discussion can be applied to pseudo differential of quantization noise. When the angular discretization width of encoder is q , the quantization error has uniform distribution with maximum error of $1/q$ and probability density of $1/q$ in every place. Since the quantization error can be assumed to be independent in every time sample

with each other, it is a white noise with $E\{\xi[i]\xi[j]\} = 0, i \neq j$ as long as S/N ratio large enough. Therefore, (23) can be formulated as:

$$E\{\xi[i]\} = 0, \quad E\{\xi[i]^2\} = \int_{-q/2}^{q/2} \frac{1}{q} x^2 dx = \frac{q^2}{12} \quad (23)$$

Here, we assume the velocity and the acceleration detection using the pseudo differential with time constant of τ .

The state equation of the pseudo differential $s/(1 + \tau s)$ driven by white noise can be derived as:

$$\dot{x} = -\frac{1}{\tau}x + \frac{1}{\tau}\xi, \quad e = -\frac{1}{\tau}x + \frac{1}{\tau}\xi \quad (24)$$

Discretized state equation is:

$$x[k+1] = e^{-\frac{T_s}{\tau}}x[k] + (1 - e^{-\frac{T_s}{\tau}})\xi[k] \quad (25a)$$

$$e[k] = -\frac{1}{\tau}x[k] + \frac{1}{\tau}\xi[k] \quad (25b)$$

Where T_s is the sampling period.

With (21),(22), and (25b), quantization noise on the pseudo differential can be formulated as:

$$\sigma\{e\}^2 = \frac{q^2 e^{\frac{T_s}{\tau}}}{6\tau^2(e^{\frac{T_s}{\tau}} + 1)} \quad (26)$$

Where T_s is sampling period. By repeating this process 2 times, quantization noise on acceleration can be obtained as:

$$\sigma\{e_{\dot{\omega}}\}^2 = \frac{q^2 e^{2\frac{T_s}{\tau}}}{3\tau^4(e^{\frac{T_s}{\tau}} + 1)^2} \quad (27)$$