

A Robot Kinematics Model Estimation Using Inertial Sensors for On-Site Building Robotics

Hiroya Sato¹, Tasuku Makabe¹, Iori Yanokura¹, Naoya Yamaguchi¹, Kei Okada¹ and Masayuki Inaba¹

Abstract—In order to make robots more useful in a variety of environments, they need to be highly portable so that they can be transported to wherever they are needed, and highly storable so that they can be stored when not in use. We propose “on-site robotics”, which uses parts procured at the location where the robot will be active, and propose a new solution to the problem of portability and storability. In this paper, as a proof of concept for on-site robotics, we describe a method for estimating the kinematic model of a robot by using inertial measurement units (IMU) sensor module on rigid links, estimating the relative orientation between modules from angular velocity, and estimating the relative position from the measurement of centrifugal force.

At the end of this paper, as an evaluation for this method, we present an experiment in which a robot made up of wooden sticks reaches a target position. In this experiment, even if the combination of the links is changed, the robot is able to reach the target position again immediately after estimation, showing that it can operate even after being reassembled. Our implementation is available on https://github.com/hiroyal224/urdf_estimation_with_imus.

I. INTRODUCTION

In today’s robotics, it is usually assumed to use well-designed robots whose dynamics and kinematics have been analyzed as accurately as possible. Therefore, when a robot is transported, the analyzed material must also be transported. Even if the robot can be disassembled or deformed, the total mass remains constant, so the larger the robot, the higher the transportation cost. In addition to transportation issues, how to store the robots is another practical problem.

One possible solution to this problem is soft robotics. The use of lightweight or shape-deformable materials can improve portability and storability. However, as is currently being actively discussed, controlling highly flexible links is difficult, and the development of new control methods to replace conventional rigid robots is a challenge.

As an alternative approach to solving the portability and storage problem, we propose a “on-site building robotics”, in which the robot is assembled with locally procured materials. This robotics improves the portability of the robot while effectively utilizing conventional rigid-body robot methods. Only the joint intermediate modules need to be kept on hand, and link parts that are no longer needed can be discarded, thereby improving storability.

However, there are many challenges to realizing such a robotics. In particular, if the robot is allowed to be composed

¹Authors are with Department of Mechano-Informatics, Graduate School of Information Science and Technology, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-8656, Japan. [h-sato, makabe, yanokura, yamaguchi, k-okada, inaba]@jsk.t.u-tokyo.ac.jp

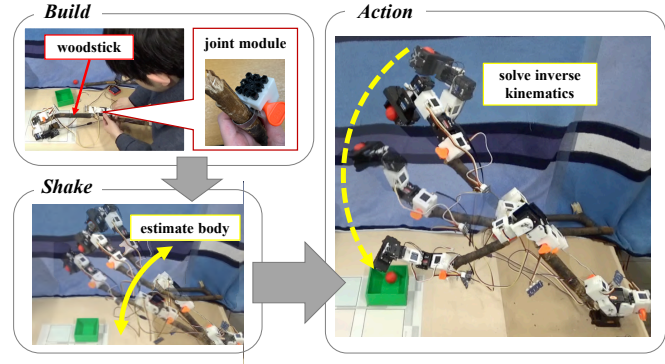


Fig. 1. Sequence of kinematic model estimation. In an experiment to evaluate our method, we used centrifugal force to estimate the length of a link by shaking a manipulator made of wooden sticks. By using the link length estimation results, the inverse kinematics of the gripper can be solved toward the target position.

of arbitrary rigid components, it must solve a hard problem of estimating its own robot model, which is necessary to realize the motion. Depending on the type of rigid rod used, it may not be straight but curved complexly. In such cases, using a ruler, for example, it is not easy to measure the relative pose between the joints, especially the relative orientation. When trying to estimate relative pose using an external camera and markers, it is still difficult because the longer the link, the harder it is to keep everything within the same field of view.

To generate a complete robot model, both kinematic and dynamic parameters are needed. According to the classical argument [1], dynamic parameters such as mass and moment of inertia can be linearly separated using matrices containing link lengths and joint angles in nonlinear form. This implies that if the relative poses of the joints and the relative angles of the joints are known, the dynamics parameters can be estimated relatively easily.

This paper focuses on the estimation of robot kinematic model using joint intermediate modules as a proof of concept for on-site building robotics. We propose a novel method for estimating the relative pose of joints using multiple inertial measurement unit (IMU) sensor modules. At the end of the paper, an evaluation experiment on a manipulator made of wooden sticks is presented to validate the effectiveness of the method.

II. RELATED WORKS

A. Soft Robotics as a Solution for Portability

Inflatable robots are being developed for their portability and ability to work in confined spaces. A classical example is the inflatable manipulator proposed by [2]. Inflatable

mobility is proposed in [3], which is a new vehicle with high portability and low impact mitigation in the event of a collision. An alternative approach is to use origami robotics [4], [5], which can be folded for compactness.

B. Link Length Estimation in Robot Calibration

Estimation of kinematic parameters of robots has been widely studied in the field of calibration [6], [7]. When the body parameters are known to some extent, methods have been proposed to correct the body parameters with small corrections, assuming that the error from the true value is small. Such methods have been used to improve the dexterity of manipulators.

In [8], [9], a method for correcting body parameters based on the pose of the target frame was proposed by defining a differential model. For multiple joint angles, the target frame calculated from the kinematic model and the actual frame pose observed by an external optical sensor are collected. The error term is then corrected by the method of least squares.

These methods assume the existence of an external optical sensor to take the true value and that the kinematic model of the robot is approximately known in advance.

C. Acquisition of Self-Body Schema and Tool Usages

In that the robots manipulate objects whose length is not known in advance, the context of tool use is similar. The use of tools by robots is often discussed in conjunction with the acquisition of a robot's self-body schema. The concept of body schema is a term in cognitive neuroscience, which originally refers to a human's self-understanding of its own body, and is often cited in robotics [10].

In [11], they proposed a learning process for a kinematic model of a robot whose model is completely unknown based on the position of AR markers attached to the links. In this study, the joint relationship between joints and links was estimated by providing a large amount of data of link pose. In [12], a neural network method was proposed to learn a tool of unknown length using information from tactile sensors.

This is similar to our work in that it uses only internal sensors. However, our proposal allows additional modules to be placed at the end of the link, and it is possible to collect data that is advantageous for estimation. We approach the problem in a simpler way that does not require learning.

D. Robot Prototyping and Modularized Robot

Robot prototyping research is usually concerned with the creation of robot components [13], [14]. In the context of educational robotics, there is an example of building a robot with an actuated paper craft capable of inductive power transfer [15]. They claimed to be useful for robot prototyping thanks to its ease of assembly. Although it is actuated by a shape memory alloy (SMA), its advanced behaviors, such as inverse kinematics, are restricted.

A modular robot is a method of constructing a robot body by repeatedly using the same parts [16]. Since the same elements are used repeatedly, a kinematic model can be obtained in principle if the connection structure is known.

However, there is a strong restriction that only one module can be used, and adding a new module requires sacrificing the advantage of repetition of the same elements.

One commercially available robot prototypable product is [17], which allows for easy robot assembly. It provides a set of easy to assemble robot bodies by combining well-designed robot modules. In contrast, our goal is to immediately assemble and operate the robot using materials available on site, which is different from the case where the components to be used are known in advance.

E. Relative Attitude Estimation of IMUs

The method of estimating relative pose of IMUs are well conducted in the field of clinical application. There have been proposed the method [18], [19] for gait analysis under the assumption that the relative position of joint (in their case, knees) with respect to IMU sensors are known. Our approach is very similar to this study, but it is not directly applicable, because their method has limitations in its use for relative pose estimation of IMUs whose relative positions are completely unknown.

III. OUR METHOD

We propose the method of estimating joints' relative pose based on the acceleration and angular velocity observed by two IMUs. An overview of our method is shown in Fig. 2.

A. Equations of Motion for IMU Sensors

If ${}^O\mathbf{r}_A$ and ${}^O\mathbf{r}_P$ denotes the position of the spatial points A , P in the 3D Euclidean space with respect to the coordinate frame O , respectively, then the following equation holds:

$${}^O\mathbf{r}_P = {}^O\mathbf{r}_A + {}^{OR_A}A\mathbf{r}_P, \quad (1)$$

where OR_A denotes the relative rotation of the coordinate frame $\{A\}$ with respect to $\{O\}$, and ${}^A\mathbf{r}_P$ the relative position of P with respect to A . By differentiate twice (1) with time, we get

$$\begin{aligned} {}^O\ddot{\mathbf{r}}_P = & {}^O\ddot{\mathbf{r}}_A + {}^{OR_A} \left([{}^A\boldsymbol{\omega}_A \times]^2 + [{}^A\dot{\boldsymbol{\omega}}_A \times] \right) {}^A\mathbf{r}_P \\ & + 2 {}^{OR_A} [{}^A\boldsymbol{\omega}_A \times] {}^A\dot{\mathbf{r}}_P + {}^{OR_A} {}^A\ddot{\mathbf{r}}_P \end{aligned} \quad (2)$$

Here we used the time derivative of rotation matrix:

$$\frac{d{}^{OR_A}}{dt} = {}^{OR_A} [{}^A\boldsymbol{\omega}_A \times]. \quad (3)$$

Now we assume that the relative motion between A and P is negligible, i.e., ${}^A\dot{\mathbf{r}}_P = {}^A\ddot{\mathbf{r}}_P = \mathbf{0}$. Then (2) reduced to

$${}^O\ddot{\mathbf{r}}_P = {}^O\ddot{\mathbf{r}}_A + {}^{OR_A} \left([{}^A\boldsymbol{\omega}_A \times]^2 + [{}^A\dot{\boldsymbol{\omega}}_A \times] \right) {}^A\mathbf{r}_P. \quad (4)$$

Letting ${}^O\mathbf{f}_A$ be a specific force, i.e., the force divide by the object's mass, acting on the origin of the frame $\{A\}$ representing by the basis of the frame $\{O\}$, then (4) becomes as follows by the Newton's law.

$${}^O\mathbf{f}_P - {}^O\mathbf{f}_A = {}^{OR_A} \left([{}^A\boldsymbol{\omega}_A \times]^2 + [{}^A\dot{\boldsymbol{\omega}}_A \times] \right) {}^A\mathbf{r}_P. \quad (5)$$

IMUs yield the specific force vector representing by the basis of their base origin. In other words, the acceleration

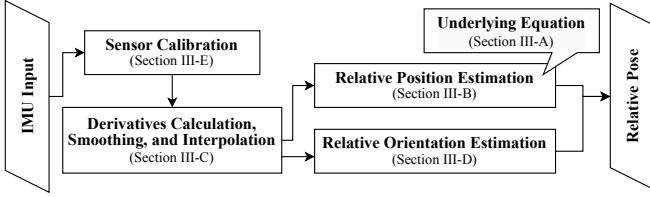


Fig. 2. Flowchart of our proposed relative pose estimation method. Corresponding sections are shown in each box.

data yielding by the IMUs are in the form ${}^A\mathbf{f}_A$ or ${}^P\mathbf{f}_P$. To obtain them, multiply from the left by ${}^A R_P = {}^O R_A^\top$.

$${}^A R_P {}^P \mathbf{f}_P - {}^A \mathbf{f}_A = \left([{}^A \boldsymbol{\omega}_A \times]^2 + [{}^A \dot{\boldsymbol{\omega}}_A \times] \right) {}^A \mathbf{r}_P. \quad (6)$$

If we already know ${}^A R_P$, the relative rotation of frame $\{P\}$ with respect to $\{A\}$, then (6) becomes the simple linear equation with unknown variable ${}^A \mathbf{r}_P$, which is what we want. We define

$$\Omega(\boldsymbol{\omega}) = [\boldsymbol{\omega} \times]^2 + [\dot{\boldsymbol{\omega}} \times]. \quad (7)$$

After an easy algebra, we get

$${}^A R_P {}^P \mathbf{f}_P - {}^A \mathbf{f}_A = {}^A R_P \Omega({}^P \boldsymbol{\omega}_P) {}^A R_P^\top {}^A \mathbf{r}_P. \quad (8)$$

Taking average with (6), we finally yield

$$\mathbf{F}({}^A \mathbf{f}_A, {}^P \mathbf{f}_P) = \bar{\Omega}({}^A \boldsymbol{\omega}_A, {}^P \boldsymbol{\omega}_P) {}^A \mathbf{r}_P, \quad (9)$$

where

$$\begin{aligned} \mathbf{F}({}^A \mathbf{f}_A, {}^P \mathbf{f}_P) &= {}^A R_P {}^P \mathbf{f}_P - {}^A \mathbf{f}_A, \\ \bar{\Omega}({}^A \boldsymbol{\omega}_A, {}^P \boldsymbol{\omega}_P) &= \frac{1}{2} (\Omega({}^A \boldsymbol{\omega}_A) + {}^A R_P \Omega({}^P \boldsymbol{\omega}_P) {}^A R_P^\top). \end{aligned} \quad (10) \quad (11)$$

B. Recursive Least Square

To achieve online estimation of ${}^A \mathbf{r}_P$, we adopt the matrix version of a recursive least square (RLS) method (see the Section 4.11.1 in [20]). We set the observed n IMU data as

$$\{({}^A \mathbf{f}_A^{(k)}, {}^P \mathbf{f}_P^{(k)}, {}^A \boldsymbol{\omega}_A^{(k)}, {}^P \boldsymbol{\omega}_P^{(k)})\}_{k=1}^n. \quad (12)$$

Let us consider the least square estimation of ${}^A \mathbf{r}_P$ based on (9). Unfortunately, because $\Omega(\boldsymbol{\omega})$ contains a second-order term of angular velocity, the mean of the residuals $\mathcal{E}\{\mathbf{F} - \bar{\Omega} {}^A \mathbf{r}_P\}$ becomes non-zero, which violates the assumption of the least square regression. To make the mean of the residuals zero, we add the correction term $K_{\bar{\Omega}}$ defined below to $\bar{\Omega}$.

$$\begin{aligned} K_{\bar{\Omega}}(\Sigma^A \boldsymbol{\omega}_A, \Sigma^P \boldsymbol{\omega}_P) &= \frac{1}{2} \left[\text{tr} \left(\Sigma^A \boldsymbol{\omega}_A + \Sigma^P \boldsymbol{\omega}_P \right) \cdot I_3 \right. \\ &\quad \left. - \left(\Sigma^A \boldsymbol{\omega}_A + {}^A R_P \Sigma^P \boldsymbol{\omega}_P {}^A R_P^\top \right) \right]. \end{aligned} \quad (13)$$

For each sample, we calculate the vector \mathbf{F} and $\bar{\Omega}_0 (= \bar{\Omega} + K_{\bar{\Omega}})$ using (10) and (11).

$$\mathbf{F}^{(k)} = {}^A R_P {}^P \mathbf{f}_P^{(k)} - {}^A \mathbf{f}_A^{(k)}, \quad (14)$$

$$\bar{\Omega}_0^{(k)} = \bar{\Omega}({}^A \boldsymbol{\omega}_A^{(k)}, {}^P \boldsymbol{\omega}_P^{(k)}) + K_{\bar{\Omega}}(\Sigma^A \boldsymbol{\omega}_A, \Sigma^P \boldsymbol{\omega}_P) \quad (15)$$

Algorithm 1 RLS for Relative Position Estimation

- 1: $P_{\text{inv}} \leftarrow \varepsilon I_3 \quad \triangleright \varepsilon > 0$ is sufficiently small value and I_n the identity matrix of size n
- 2: ${}^A \hat{\mathbf{r}}_P \leftarrow \mathbf{0}_3 = (0, 0, 0)^\top$
- 3: Set $\Sigma^A \boldsymbol{\omega}_A, \Sigma^P \boldsymbol{\omega}_P \quad \triangleright$ obtain during the calibration step described in Section III-E
- 4: $\text{Diffs} = [\mathbf{0}_3, \dots, \mathbf{0}_3] \quad \triangleright$ queue of length N_{lag}
- 5: **function** RLS(${}^A \mathbf{f}_A, {}^P \mathbf{f}_P, {}^A \boldsymbol{\omega}_A, {}^P \boldsymbol{\omega}_P; {}^A \hat{\mathbf{r}}_P, {}^A R_P$)
- 6: Calculate $\mathbf{F}, \bar{\Omega}_0 \quad \triangleright$ Eqs. (14), (15)
- 7: Enqueue $\mathbf{F} - \bar{\Omega}_0 {}^A \hat{\mathbf{r}}_P$ to Diffs
- 8: Dequeue the front entry of Diffs
- 9: $C \leftarrow (\Sigma_{\delta \in \text{Diffs}} \delta \delta^\top) / (N_{\text{lag}} - 1) \quad \triangleright$ Eq. (20)
- 10: $P_{\text{inv}} \leftarrow P_{\text{inv}} + \gamma_{\text{pos}} \cdot \bar{\Omega}_0^\top C^{-1} \bar{\Omega}_0 \quad \triangleright$ Eq. (18)
- 11: $\mathbf{Q} \leftarrow \mathbf{Q} + \gamma_{\text{pos}} \cdot \bar{\Omega}_0^\top C^{-1} \mathbf{F} \quad \triangleright$ Eq. (19)
- 12: ${}^A \hat{\mathbf{r}}_P \leftarrow P_{\text{inv}}^{-1} \mathbf{Q}, \quad \hat{\Sigma}^A \mathbf{r}_P \leftarrow P_{\text{inv}}^{-1}$
- 13: **return** ${}^A \hat{\mathbf{r}}_P, \hat{\Sigma}^A \mathbf{r}_P$
- 14: **end function**
- 15: **while** sensors are available **do**
- 16: ${}^A \mathbf{f}_A, {}^P \mathbf{f}_P, {}^A \boldsymbol{\omega}_A, {}^P \boldsymbol{\omega}_P \leftarrow \text{CALIBRATED_IMU_DATA}$
- 17: ${}^A R_P \leftarrow \text{mode of } \mathfrak{B}(A) \quad \triangleright$ Eq. (36) in Section III-D
- 18: ${}^A \hat{\mathbf{r}}_P, \hat{\Sigma}^A \mathbf{r}_P \leftarrow \text{RLS}({}^A \mathbf{f}_A, {}^P \mathbf{f}_P, {}^A \boldsymbol{\omega}_A, {}^P \boldsymbol{\omega}_P; {}^A \hat{\mathbf{r}}_P, {}^A R_P)$
- 19: **end while**

Now we try to find the minimizer ${}^A \mathbf{r}_P$ of $J[n]$ defined by the following:

$$J[n] = \frac{1}{2} \sum_{k=0}^n \gamma_{\text{pos}}^{n-k} \left\| \mathbf{F}^{(k)} - \bar{\Omega}_0^{(k)} {}^A \mathbf{r}_P \right\|^2, \quad (16)$$

where $\gamma_{\text{pos}} \in (0, 1]$ is the forgetting factor. The optimized solution ${}^A \hat{\mathbf{r}}_P[n]$ that minimizes $J[n]$ is written as follows.

$${}^A \hat{\mathbf{r}}_P[n] = P[n] \mathbf{Q}[n], \quad (17)$$

where

$$P[n]^{-1} = P[n-1]^{-1} + \gamma_{\text{pos}} \cdot \bar{\Omega}_0^{(n)\top} C[n]^{-1} \bar{\Omega}_0^{(n)}, \quad (18)$$

$$\mathbf{Q}[n] = \mathbf{Q}[n-1] + \gamma_{\text{pos}} \cdot \bar{\Omega}_0^{(n)\top} C[n]^{-1} \mathbf{F}^{(n)}. \quad (19)$$

Letting $\boldsymbol{\delta}^{(i)}[k] = \mathbf{F}^{(i)} - \bar{\Omega}_0^{(i)} {}^A \hat{\mathbf{r}}_P[k]$, we define $C[n]$ as below.

$$C[n] = \frac{1}{N_{\text{lag}} - 1} \sum_{i=n-N_{\text{lag}}}^n \boldsymbol{\delta}^{(i)}[n-1] \boldsymbol{\delta}^{(i)}[n-1]^\top, \quad (20)$$

where $\boldsymbol{\delta}^{(i)}$ is set to zero if $i < 0$. N_{lag} , a sample time of estimation error, is an arbitrary positive integer larger than two. We set $N_{\text{lag}} = 100$. Note that if the number of non-zero $\boldsymbol{\delta}^{(i)}$ in the summation is less than 3, then $C[n]$ is always singular matrix. In this case, we use Moore-Penrose pseudoinverse instead of the ordinal inverse $C[n]^{-1}$. This step is needed only in the beginning of the estimation.

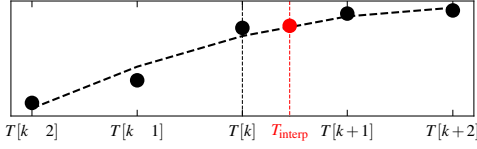


Fig. 3. Time interpolation using Savitzky-Golay filter with $N = 2$. If we want to align with the value at time T_{interp} , we can use polynomial interpolation to estimate the value at that time. If T_{interp} falls outside the interpolation range, the closer of the values of $T[k \pm N]$ is used instead.

C. Savitzky-Golay Filter

To estimate the relative position of IMUs from the centrifugal force, it is necessary to obtain a term for the time derivative of the angular velocity. We calculated the time derivative using a Savitzky-Golay filter [21]. It is achieved by least-squares fitting of a polynomial of degree M to the local $2N + 1$ points, where M and N are hyperparameters to be determined. We use $M = 5$ and $2N + 1 = 7$ in the experiments. Details are discussed in Section V-A.

We now consider applying a filter to the observed data sequence $\{(T[i], d[i])\}_i$. The symbol d is substituted for each of the acceleration and angular velocity terms. The approximated polynomial can be written as follows.

$$p[k](t) = \sum_{i=0}^M c_{k,i} (t - T[k])^i, \quad (21)$$

where $c_{k,i}$ ($i = 0, \dots, M$) is the least square solution of

$$\begin{bmatrix} (\Delta T_{k-N})^0 & \cdots & (\Delta T_{k-N})^M \\ \vdots & \ddots & \vdots \\ (\Delta T_{k+N})^0 & \cdots & (\Delta T_{k+N})^M \end{bmatrix} \begin{bmatrix} c_{k,0} \\ \vdots \\ c_{k,M} \end{bmatrix} = \begin{bmatrix} d[k-N] \\ \vdots \\ d[k+N] \end{bmatrix}. \quad (22)$$

Here we set $\Delta T_{k-j} := T[k-j] - T[k]$ and $(\Delta T_k)^0 = 1$. We can easily calculate the least square solution by $\hat{c} = (A^T A)^{-1} A^T \mathbf{d}$ where A is the $(2N + 1) \times M$ coefficient matrix of $(c_{k,0}, \dots, c_{k,M})^T$ defined in (22) and $\mathbf{d} = (d[k-N], \dots, d[k+N])^T$. Note that $A^T A$ is invertible if the sampling timestamp $T[k+j]$ is different for all $j = -N, \dots, N$, because A contains $M \times M$ Vandermonde's matrix as a submatrix and its determinant becomes non-zero. In the ordinal Savitzky-Golay filter, it is usually assumed that $\Delta T_{k+j} = j$ for all $j = -N, \dots, N$ to reduce a computational cost. In our implementation, we decided to recalculate timesteps to handle with sudden change of sampling rate due to communication issues. To obtain the derivative of ω , we just differentiate the interpolating polynomial.

$$\frac{dp}{dt}[k] = \sum_{i=1}^M i \cdot c_{k,i} (t - T[k])^{i-1}, \quad (23)$$

where $c_{k,i}$ is the coefficient of the interpolating polynomial.

If \mathbf{d} is multidimensional, it can be computed at once by replacing the coefficient vector \mathbf{c} with a matrix C . For example, if we set $\mathbf{d} = (\omega^T, \mathbf{a}^T)^T$, C can be computed as

$$C = (A^T A)^{-1} A^T \begin{bmatrix} \omega[k-N]^T & \mathbf{a}[k-N]^T \\ \vdots & \vdots \\ \omega[k+N]^T & \mathbf{a}[k+N]^T \end{bmatrix}. \quad (24)$$

Each of column of C corresponds to the each of data column.

Furthermore, thanks to polynomial interpolation, it is possible to match the timestamps with other sensor modules. Synchronizing the time among sensors is very important in our method, which focuses on the acceleration and angular velocity values at the moment. We assume here that the each sensor's wall clock is well calibrated, e.g., by using an NTP server, and that only the timestamps of the data are not the same. The time alignment is illustrated in Fig. 3.

D. Bingham Filter for Relative Orientation Estimation

1) *Principle of Relative Orientation Estimation:* Now we assume that the relative rotation ${}^A R_P$ is also unknown. Assuming that the link is rigid, we can use the classical rigid body theorem that the angular velocity on the body is all the same. If i -th IMU and j -th IMU are on the same rigid body, then the following equation holds:

$${}^i \omega_i = {}^i R_j {}^j \omega_j. \quad (25)$$

Now we can estimate ${}^i R_j$ from this equation.

2) *Quaternion and Bingham Distribution:* We try to estimate the probability distribution of ${}^i R_j$. To realize this, we made use of the idea of the Bingham distribution-based linear filter proposed by [22]. The Bingham distribution is the probability distribution over the unit (hyper)-sphere.

The Bingham distribution over 3-sphere is defined as

$$\mathfrak{B}(A)(\mathbf{q}) \propto \exp(\mathbf{q}^T A \mathbf{q}), \quad (26)$$

where $\mathbf{q} \in \mathbb{R}^4$ is a unit quaternion and $A \in \mathbb{R}^{4 \times 4}$ is a parameter of distribution. Now we define the notation here.

- 1) The set of all unit quaternion is denoted as $\mathbb{S}^3 \subset \mathbb{R}^4$.
- 2) We use "wxyz" notation.
- 3) While q is a quaternion, \mathbf{q} is the vector in \mathbb{R}^4 in the form of $(w, x, y, z)^T$, where w is a scalar part of q .
- 4) The *conjugate* of quaternion q is written as q^* .
- 5) The quaternion product is denoted as \odot .

Letting $\mathbf{v} = (x, y, z)^T \in \mathbb{R}^3$, we define a tuple notation of quaternion as

$$q = \begin{pmatrix} w \\ \mathbf{v} \end{pmatrix} = \begin{pmatrix} w \\ (x, y, z)^T \end{pmatrix} = (w, \mathbf{v})^T. \quad (27)$$

Now we define the multiplication of quaternions \odot as follow:

$$\begin{pmatrix} w \\ \mathbf{v} \end{pmatrix} \odot \begin{pmatrix} w' \\ \mathbf{v}' \end{pmatrix} := \begin{pmatrix} ww' - \mathbf{v}^T \mathbf{v}' \\ w\mathbf{v}' + w'\mathbf{v} + \mathbf{v} \times \mathbf{v}' \end{pmatrix}. \quad (28)$$

It is well known that quaternions can be express the 3D spatial rotation [23]. If we set the conversion function from unit quaternion to rotation matrix $R: \mathbb{S}^3 \rightarrow \text{SO}(3)$, we get

$$\mathbf{q} \odot (0, \mathbf{x})^T \odot \mathbf{q}^* = (0, R(\mathbf{q})\mathbf{x})^T \quad (29)$$

3) *Bingham Linear Filter:* From (29), we can yield

$$\mathbf{q} \odot (0, \mathbf{b})^T = (0, \mathbf{a})^T \odot \mathbf{q}, \quad (30)$$

where $\mathbf{a} \in \mathbb{R}^3$ is a point after the rotation and $\mathbf{b} \in \mathbb{R}^3$ a point before the rotation. Using (28), we get the following.

$$\underbrace{\begin{bmatrix} 0 & (\mathbf{a} - \mathbf{b})^T \\ -(\mathbf{a} - \mathbf{b}) & [(\mathbf{a} + \mathbf{b}) \times] \end{bmatrix}}_{H(\mathbf{a}, \mathbf{b})} \mathbf{q} = \mathbf{0}. \quad (31)$$

Note that all \mathbf{q} that satisfy (31) also satisfy $\mathbf{a} = R(\mathbf{q})\mathbf{b}$ and vice versa.

The above discussion was for the case where both \mathbf{a} and \mathbf{b} have no noise. In practice, there is observational noise, so we need to consider it. Based on [22], separating the noise term from the true value term, then we have

$$H(\mathbf{a} + \varepsilon_a, \mathbf{b} + \varepsilon_b)\mathbf{q} = H(\mathbf{a}, \mathbf{b})\mathbf{q} + \varepsilon_H. \quad (32)$$

The mean of ε_H is zero. Under the assumption that ε_a and ε_b are uncorrelated, we can compute its covariance matrix Σ^H by

$$\Sigma^H = N \left(\begin{bmatrix} \varepsilon_a \\ \varepsilon_b \end{bmatrix} \otimes \Sigma^q \right) N^\top, \quad (33)$$

where N is a 4×24 matrix defined by

$$[H(\mathbf{e}_1, \mathbf{0}) \ H(\mathbf{e}_2, \mathbf{0}) \ H(\mathbf{e}_3, \mathbf{0}) \ H(\mathbf{0}, \mathbf{e}_1) \ H(\mathbf{0}, \mathbf{e}_2) \ H(\mathbf{0}, \mathbf{e}_3)]. \quad (34)$$

Here \mathbf{e}_i is a vector whose i -th element is 1 and others 0. The matrix Σ^q is the covariance matrix of rotation. We utilize for Σ^q the covariance of the uniform distribution over the entire quaternion space \mathbb{S}^3 . It can be written as $\Sigma^q = 0.25I_4$, where I_4 is the identity matrix of size 4.

Now we formulate the sequential update formula of the Bingham distribution. First, calculate

$$\Delta A_{i,j} = -\frac{1}{2} H(\boldsymbol{\omega}_i, \boldsymbol{\omega}_j)^\top (\Sigma^H)^{-1} H(\boldsymbol{\omega}_i, \boldsymbol{\omega}_j) \quad (35)$$

to obtain an updater. If one assumes Σ^e is fixed during the task, then $(\Sigma^H)^{-1}$ can be precomputed and one can save the computational cost.

Letting $\gamma_{\text{rot}} \in (0, 1]$ be a forgetting factor, we can update the parameter matrix with $\Delta A_{i,j}$ as

$$A_{i,j}[n+1] = \gamma_{\text{rot}} \cdot A_{i,j}[n] + \Delta A_{i,j}. \quad (36)$$

Here $A_{i,j}$ is the parameter matrix of the Bingham distribution of iR_j and initialized with zero matrix: $A_{i,j}[0] = O_{4 \times 4}$.

We can extract the mode quaternion by eigendecomposition. The eigenvector of $A[n]$ corresponding to the maximum eigenvalue is the mode quaternion of the Bingham distribution defined by $\mathfrak{B}(A[n])$. We use the mode rotation as iR_j .

The approach to pose estimation using the Bingham distribution does not require a linearization operation and the update rule is simple. In addition, it has the advantage of handling global information over the entire rotation space.

E. Brief Calibration of IMU sensors

Before estimating the relative posture of the joints, a simple calibration of the IMU was performed. This is an important operation because the centrifugal force value is critical in this method.

This process is performed as follows. First, the robot is kept stationary and approximately 1000 samples are collected. During the stationary state, the bias of the gyro sensor and the covariance between the gyro sensor, its derivative, and the acceleration are measured. The bias and covariance of the sensors were assumed to be constant during operation, independent of angular velocity or acceleration, and these values were used as constants.

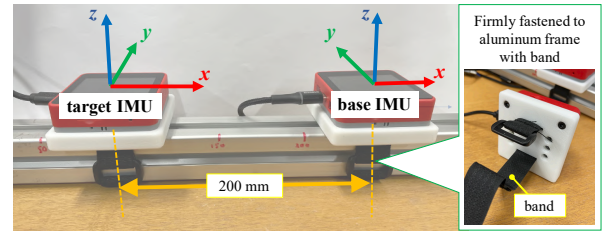


Fig. 4. Setup of the validation experiment. Two M5Stack Fire were used as the IMU modules. The IMU modules were fixed to aluminum rods with 200 mm spacing on the rods.

Next, while we slowly rotated the robot, approximately 10000 samples were observed and used to calibrate the accelerometers. Calibration of the accelerometers is performed by the ellipsoid fitting method, as presented in [24]. This method is based on the assumption that the magnitude of the gravitational acceleration is constant independent of the orientation of the IMU. The reason for turning the robot slowly is to ensure that, as far as possible, acceleration other than gravitational acceleration is not observed. If the robot is fixed to the environment and difficult to rotate, pre-calibrated IMU sensors can be used.

After collecting the data, the calibration results are obtained by the least-squares method. If we observe an acceleration dataset $\{(X_i, Y_i, Z_i)\}_{i=1}^N$, these points should be on the ellipsoid represented by

$$aX^2 + bXY + cXZ + dY^2 + eYZ + fZ^2 + pX + qY + rZ + s = 0 \quad (37)$$

where a, \dots, f, p, \dots, s are real numbers. With a dataset of observation, we can construct linear square least problem that find a, \dots, f, p, \dots, s minimizing

$$\underbrace{\begin{bmatrix} X_1^2 & X_1Y_1 & X_1Z_1 & Y_1^2 & Y_1Z_1 & Z_1^2 & X & Y & Z & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_N^2 & X_NY_N & X_NZ_N & Y_N^2 & Y_NZ_N & Z_N^2 & X & Y & Z & 1 \end{bmatrix}}_{D_{1:N}} \underbrace{\begin{bmatrix} a \\ \vdots \\ s \end{bmatrix}}_v \quad (38)$$

Now we have the optimal parameter of the quadratic cone. However, it does not necessarily ellipsoid. In practice, we restricted our estimation results to ellipsoid surfaces that are not hyperbolic or parabolic, based on [25].

IV. EXPERIMENTS

A. Validation Experiment

As a validation experiment, we used a pair of IMU sensor modules strapped to an aluminum rod of known length. Fig. 4 shows the setup used in the evaluation experiment. The modules were aligned to the same orientation. To verify that the relative pose was estimated correctly, we shook the rods by hand. In this section, we set $\gamma_{\text{pos}} = \gamma_{\text{rot}} = 1$.

Fig. 5 shows the estimation results. The position estimation error is at most about 3 mm and the orientation error is at most about 3° , which are good results. The square root of the trace of the covariance matrix of the estimation results is about 1 mm at the end. By analogy with the normal

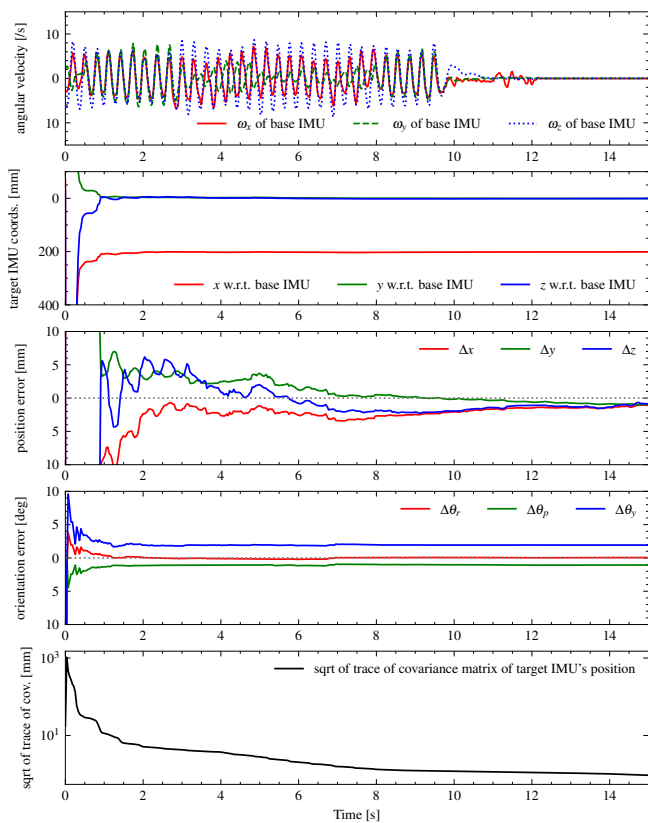


Fig. 5. Results of validation experiments. From the top row, the time variation of the angular velocity during the experiment, the estimated relative position, the position error, the orientation error, and the square root of the trace of covariance of the estimation results are depicted.

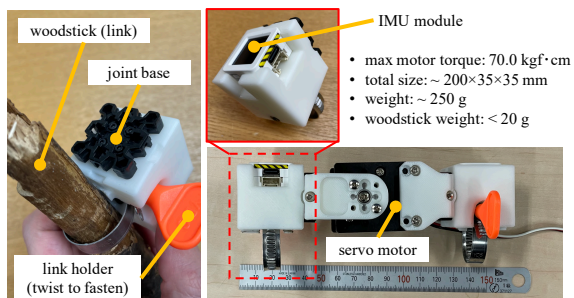


Fig. 6. Our joint module used in the application experiments.

distribution, three times this value should be approximately the 99.7% percentile, but this value is seemed to be an underestimate considering the actual error. One possible reason for the underestimation is that the assumption of least-squares regression that the error distribution is Gaussian is not strictly met. Although the assumption is not being followed, we formally utilize the trace of the covariance matrix as a termination condition for the estimation.

B. Design of the Joint Module Used in the Application Experiment

We developed a joint intermediate module for on-site built robots that have:

- 1) a structure to fix the link and the module,
- 2) an actuator (e.g. servo motors),

- 3) two IMU sensors.

Requirement 3 is needed to simplify the estimation problem. One of the two sensors is fixed to the servo motor and the other to the servo horn (child side of the drive shaft). This arrangement allows the two IMU sensors to be mounted on the same rigid link and the proposed method can be used.

The appearance and specifications of the joint intermediate module are shown in Fig. 6. The module consists of joint parts, a pair of IMU modules, a servo motor, and 3D printed structures. The motor on the root side has enough power to drive the entire arm. A small, lightweight motor is used on the tip side to reduce the motor torque on the root side.

Since the position of the IMU with respect to the joint module is known at the design stage, the relative position of the IMUs relative to the joint can be given in advance. Using this information, the relative pose between the joints can be calculated from the relative pose between the IMUs.

C. Application Experiment

We show that a manipulator made of wooden sticks can reach a gripper to a target position with our estimation method. The wooden stick used was a cherry tree (Someiyoshino), which was picked up by the author on our university campus. Reaching was assumed successful by dropping a red ball ($\phi 20$ mm) into a placed green box (90 mm \times 90 mm \times 40 mm). The position of the green box was given in advance relative to the robot's reference coordinates. No visual feedback was used, and only the estimated link length was used for the inverse kinematics of the gripper.

The experiment is shown in Fig. 7 and the estimation result Fig. 8. As soon as the estimation motion started ($T = 0$), a signal was sent to start estimation and the previous estimation result was reset. The sampling rate of IMUs was 85 Hz.

A swinging signal was sent to a randomly selected joint at a fixed angle amplitude. The estimation process stopped automatically ($T = 47$) when the covariance fell below a threshold, which was set at 5 mm at “the 95th percentile”, and the orientation at 0.1 deg at “the 95th percentile”.

When the estimation motion was completed, the robot reached the gripper to the target box based on the estimated kinematic model. Note that although the actual robot deflected due to gravity, this effect was assumed to be negligible in this experiment because only the horizontal plane was focused on in this experiment. To prevent collision with the box due to deflection, the offset was set 50 mm above the box. When the gripper reached the target position, it automatically opened and dropped the ball ($T = 63$). As shown in Fig. 7, the robot successfully dropped the red ball into the green box.

Next, the servo motors of the robot were turned off, the robot was disassembled, and the configuration was changed. We replaced the wooden sticks with different ones and changed the mounting direction drastically of the joint modules. At this time, the robot was manually moved like a puppet to make sure the gripper could reach the green box.

The motors were turned on again and estimation was started at time $T = 250$. The estimation was completed ($T =$

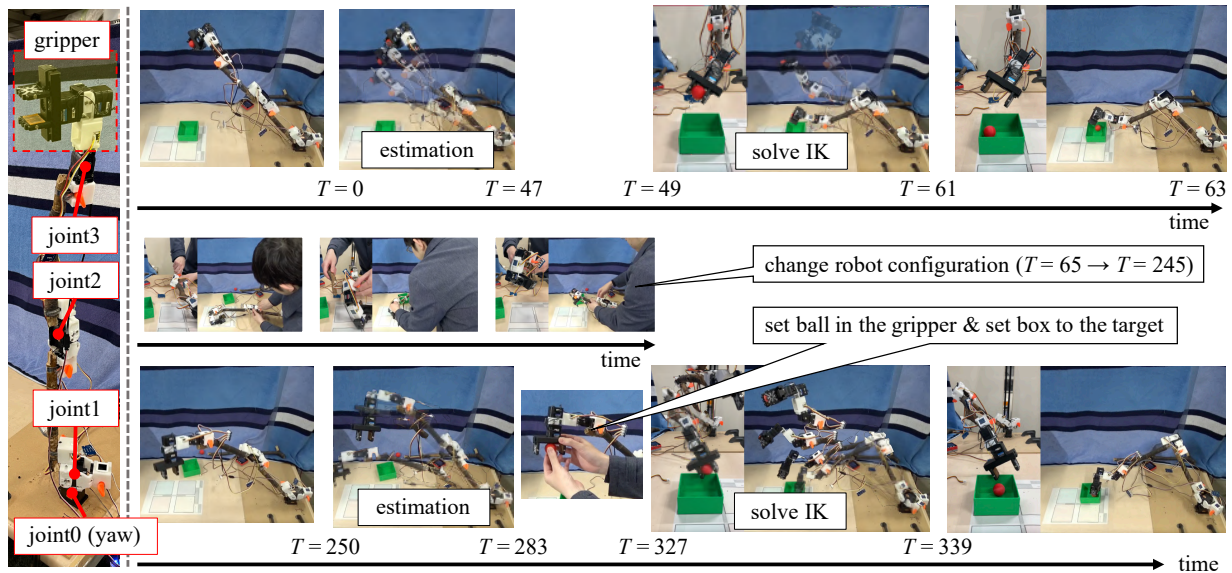


Fig. 7. Snapshot of an application experiment. The reaching motion is considered successful if the red ball can be dropped into the green box. Although the configuration of the robot was changed during $T = 65 \rightarrow T = 245$, reaching was successful both before and after the change. The joint configuration of the robot is illustrated on the left side of the figure. “joint1” to “joint3” can be rotated in the pitch direction.

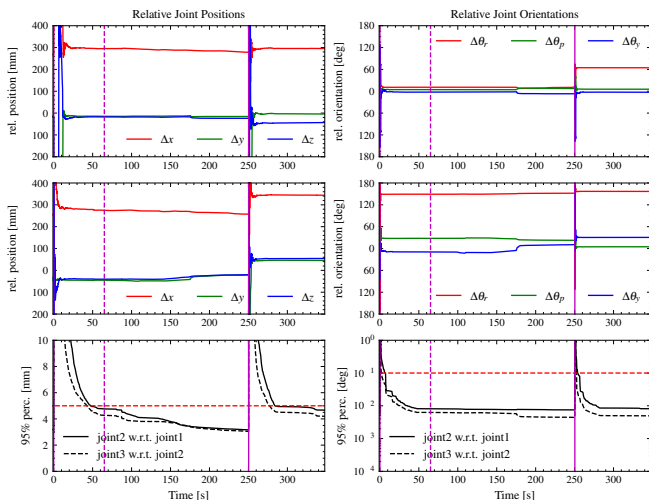


Fig. 8. Estimation results during an application experiment. From the top row, time variation of the relative pose of joint1 to joint2, the relative pose of joint2 to joint3, and the 95% percentile (twice the square root of the trace of the covariance matrices) the pose variance. The vertical dashed line in magenta is the time when the robot started to change its configuration. The solid magenta vertical line is the time when the estimated motion began.

283) and the gripper moved over the target box ($T = 339$) and successfully dropped the ball into the box. Comparing the final posture of the robot when the ball was dropped, it can be seen that the angles of each joint were very different. This indicates that the robot solved the inverse kinematics again based on the estimated joints’ relative poses.

V. DISCUSSIONS

A. Effect of Sampling Frequency of IMUs

Our method uses a Savitzky-Golay filter to smooth and interpolate the sensor data and calculate the derivative values. There is concern that this frequency characteristics may

affect the estimated values. The characteristics is determined by the ratio of the shake frequency to the sampling frequency.

To discuss the effect of sampling frequency on estimation, simulations were performed assuming an IMU sensor placed at relative position $(L, 0, 0)$ and rotated with angular velocity $(0, 0, \sin(2\pi ft))$, where f is a frequency of shaking and t is the time. No Gaussian noise is added here. Throughout the simulation, it was observed that the ratio of the estimated relative position \hat{L} to the true value L and the ratio to swing frequency f to sampling frequency f_{sample} were constant for the magnitudes of L and f . Based on this observation, we considered the relationship between the non-dimensionalized values of \hat{L}/L and f/f_{sample} . In the simulation, the sampling frequency was 100 Hz.

Fig. 9 shows the simulation results. The red dashed line labeled “-3dB border” indicates a line whose value is $10^{-3/20} \approx 0.71$. This value follows the fact that the filtering cutoff gain is usually defined at -3dB. The results show that the ratio \hat{L}/L worsens as the ratio f/f_{sample} increases. For $N = 3$ and $M = 5$, the ratio \hat{L}/L falls below the line around $f/f_{\text{sample}} \approx 0.3$. This suggests that too fast vibration can cause estimation accuracy to deteriorate.

It is important to note that the way in which this change occurs and the value of the cutoff frequency will differ greatly depending on how M and N are determined. However, in all cases, the higher the frequency, the lower the performance of filtering. For a general discussion of the frequency response of the Savitzky-Golay filter, see [21].

B. Effect of Estimation Motion

Let us consider how the estimation motion affect the estimation results. Noticing that $K_{\bar{\Omega}}(\Sigma^A \omega_A, \Sigma^P \omega_P)$ in (15) is constant if we assume the covariances of sensor noise are unchange during operation,

$$\bar{\Omega}_0^{(k)} \approx \bar{\Omega}(A \omega_A^{(k)}, P \omega_P^{(k)}) \quad (39)$$

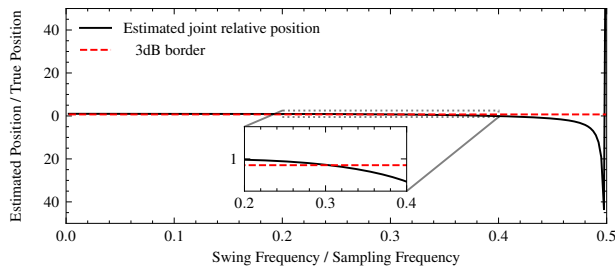


Fig. 9. Frequency characteristics of our filter. In our experiments, we set the number of points in the window $N = 3$ and polynomial degree $M = 5$.

for a large angular velocity. According to (18), we can see $P[n]^{-1}$ tends to have large entries as $\bar{\Omega}_0^{(k)}$ increases. This suggests that the more intense the shaking, the smaller the estimation variance due to the error term is expected to be.

C. Limitation and Future Works

In our method, we use a pair of IMU sensors installed on the same rigid link to estimate the relative pose between the IMUs. If one tries to estimate the pose of the end effector directly, the entire robot may vibrate, and the rigid body assumption may not be satisfied. In the verification experiments, we estimated the length of each link one by one. However, even in cases where the rigid body assumption fails locally, it would be possible to further improve performance by making full use of sensor information, such as by applying a graph-based SLAM method [26].

In this paper, we focused on the estimation of the kinematic model. In order to generate the behavior of more complex robots, a dynamic model should be required, whose estimation is our future work.

VI. CONCLUSION

In this paper, we proposed a method for estimating the kinematic model of a robot composed of arbitrary rigid links using an IMU sensor module for proof of concept of on-site building robotics. We created a manipulator made up of wooden sticks using the designed joint intermediate module. Even after making significant changes to the robot's configuration, we confirmed that the robot was able to reach the object successfully, which suggests that the robot was able to correctly estimate its kinematic model by our method.

REFERENCES

- [1] C. G. Atkeson, C. H. An, and J. M. Hollerbach, "Estimation of inertial parameters of manipulator loads and links," *The International Journal of Robotics Research*, vol. 5, no. 3, pp. 101–119, 1986. [Online]. Available: <https://doi.org/10.1177/027836498600500306>
- [2] M. Rybski, M. Shoham, and G. Grossman, "Robotic manipulators based on inflatable structures," *Robotics and Computer-Integrated Manufacturing*, vol. 12, no. 1, pp. 111–120, 1996. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0736584595000224>
- [3] H. Sato, Y. a. Seong, R. Yamamura, H. Hayashi, K. Hata, H. Ogata, R. Niiyama, and Y. Kawahara, "Soft yet strong inflatable structures for a foldable and portable mobility," in *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, ser. CHI EA '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–4. [Online]. Available: <https://doi.org/10.1145/3334480.3383147>

- [4] J. Santoso and C. D. Onal, "An origami continuum robot capable of precise motion through torsionally stiff body and smooth inverse kinematics," *Soft Robotics*, vol. 8, no. 4, pp. 371–386, 2021, pMID: 32721270. [Online]. Available: <https://doi.org/10.1089/soro.2020.0026>
- [5] C. Sanprueksin and C. Wongviriyawong, "Kinematics and design of a rigid foldable origami robot," in *2017 18th International Conference on Advanced Robotics (ICAR)*, 2017, pp. 161–166.
- [6] W. Veitschegger and C.-H. Wu, "Robot calibration and compensation," *IEEE Journal on Robotics and Automation*, vol. 4, no. 6, pp. 643–656, 1988.
- [7] B. Mooring, Z. Roth, and M. Driels, *Fundamentals of Manipulator Calibration*. Wiley, 1991. [Online]. Available: <https://books.google.co.jp/books?id=jD5SAAAAMAAJ>
- [8] S. H. Ye, Y. Wang, Y. J. Ren, and D. K. Li, "Robot calibration using iteration and differential kinematics," *Journal of Physics: Conference Series*, vol. 48, no. 1, p. 1, oct 2006. [Online]. Available: <https://dx.doi.org/10.1088/1742-6596/48/1/001>
- [9] H. M. Balanji, A. E. Turgut, and L. T. Tunc, "A novel vision-based calibration framework for industrial robotic manipulators," *Robotics and Computer-Integrated Manufacturing*, vol. 73, p. 102248, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0736584521001289>
- [10] M. Hoffmann, H. Marques, A. Hernandez Arieta, H. Sumioka, M. Lungarella, and R. Pfeifer, "Body schema in robotics: A review," *Autonomous Mental Development, IEEE Transactions on*, vol. 2, pp. 304 – 324, 10 2010.
- [11] J. Sturm, C. Plagemann, and W. Burgard, *Body Schema Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 131–161. [Online]. Available: https://doi.org/10.1007/978-3-642-25116-0_12
- [12] Y. K. Cota Nabeshima and M. Lungarella, "Adaptive body schema for robotic tool-use," *Advanced Robotics*, vol. 20, no. 10, pp. 1105–1126, 2006. [Online]. Available: <https://doi.org/10.1163/156855306778522550>
- [13] I. Ebert-Uphoff, C. Gosselin, D. Rosen, and T. Laliberte, *Rapid Prototyping for Robotics*, 07 2005.
- [14] K. Rajan, M. Samykan, K. Kadirgama, W. S. W. Harun, and M. M. Rahman, "Fused deposition modeling: process, materials, parameters, properties, and applications," *The International Journal of Advanced Manufacturing Technology*, vol. 120, no. 3, pp. 1531–1570, 2022.
- [15] Z. Kening and S. Zhao, "Autogami: A low-cost rapid prototyping toolkit for automated movable paper craft," 04 2013, pp. 661–670.
- [16] K. Gilpin and D. Rus, "Modular robot systems," *IEEE Robotics & Automation Magazine*, vol. 17, no. 3, pp. 38–55, 2010.
- [17] Studica, "What is robot prototyping?" 2023, march 15, 2024. [Online]. Available: <https://www.studica.com/blog/robot-prototyping/>
- [18] M. Kok, K. Eckhoff, I. Weygers, and T. Seel, "Observability of the relative motion from inertial data in kinematic chains," *Control Engineering Practice*, vol. 125, p. 105206, 08 2022.
- [19] I. Weygers, M. Kok, H. De Vroey, T. Verbeest, M. Versteyshe, H. Hallez, and K. Claeys, "Drift-free inertial sensor-based joint kinematics for long-term arbitrary movements," *IEEE Sensors Journal*, vol. 20, no. 14, pp. 7969–7979, 2020.
- [20] T. Moon and W. Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Prentice Hall, 2000.
- [21] R. W. Schafer, "On the frequency-domain properties of savitzky-golay filters," in *2011 Digital Signal Processing and Signal Processing Education Meeting (DSP/SPE)*, 2011, pp. 54–59.
- [22] R. A. Srivatsan, M. Xu, N. Zevallos, and H. Choset, "Probabilistic pose estimation using a bingham distribution-based linear filter," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1610–1631, 2018. [Online]. Available: <https://doi.org/10.1177/0278364918778353>
- [23] J. Gallier, "The Quaternions and the Spaces S^3 , $SU(2)$, $SO(3)$, and $\mathbb{R}P^3$," in *Geometric Methods and Applications: For Computer Science and Engineering*, 2011, ch. 9, pp. 281–300.
- [24] Y. Sun and X. Xu, "Calibration of mems triaxial accelerometers based on the maximum likelihood estimation method," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–10, 09 2020.
- [25] Q. Li and J. Griffiths, "Least squares ellipsoid specific fitting," in *Geometric Modeling and Processing, 2004. Proceedings, 2004*, pp. 335–340.
- [26] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.