

Motion Locus Analysis to Detect Rotation

Toshiki Iso

Abstract—For mobile applications such as command interfaces and canine healthcare, we propose a method that can detect rotation by analyzing the motion locus calculated from motion sensor data such as three axis angular velocities. While conventional methods demand the use of multiple sensors, our proposal uses only a single gyro-sensor. The method judges rotation by setting criteria based on the distortion of normal vectors extracted from the motion locus. The distortion of normal vectors is defined by principle component analysis. Therefore, it has low computation costs. We evaluate two variants of the proposed method in a comparison of normal vector extraction and confirm their superior performance. The results show the method can be easily implemented on not so high-spec mobile devices.

Index Terms—Principal component analysis, rotational motion, motion locus, cross product, plane approximation.

I. INTRODUCTION

Recent mobile phones come in a variety of forms and are used in situations such as wearable devices and IoT (Internet of Things), as they have rich functions associated with communication and sensing. We need, however, to improve the interfaces to better match them to their usage scenarios. For example, wrist-watch type phones [1] are so small that key-operations become difficult. A convenient solution is to use a gesture command interface such as rolling the user's wrist. Another example is the smart dog collar for canine healthcare [2]. Continuous head or body scratching is an indicator of disease. Setting motion sensors on the collar makes it possible to automatically detect the dog's actions (see Fig. 1). These services require simple devices and algorithms for detecting rotational motions.

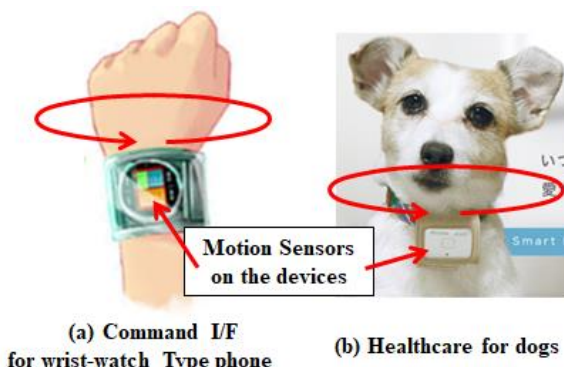


Fig. 1. Service scenes of detecting rotational motions.

Few conventional methods have been designed to detect rotational motions by using motion sensors. Most studies on recognizing human activities treat only acceleration data

[3]-[8]. Thus, they can detect only transition motions and have limited ability to detect rotational motions. (“rotation” is defined here to contain not only circles but also arcs.) A gyro-sensor (angular velocity sensor) is basically used for detecting rotational target object movement in three dimensions, however, no trial has demonstrated the detection of rotational motion. Because the gyro-sensor is mounted on the target object, it is possible to calculate motion locus in the sensor coordinate system. Therefore, we attempt to detect rotation of the target object by analyzing motion locus from just gyro-sensor data.

II. PROPOSED APPROACH

When a target object is experiencing rotational motion, the motion locus should contain the plane of rotation. In this situation, if normal vectors can be extracted, we can judge whether the rotational motion is stable by analyzing their deviations. Our method has four stages; the first calculates motion locus, the second detect candidates of normal vectors, the third uses principle component analysis to detect the representative normal vectors, the final stage calculates the stability of normal vector deviations (see Fig. 2). We explain each stage below using the example of a wrist-watch type phone.

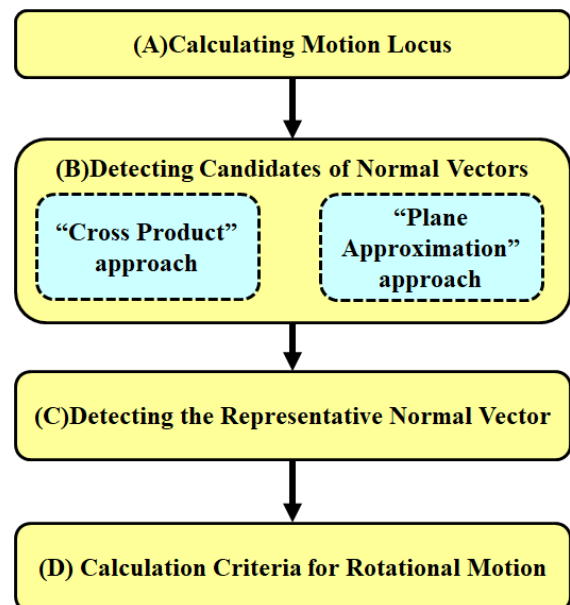


Fig. 2. Flow of proposed method.

A. Calculating Motion Locus

A wrist-watch type phone has motion sensors such as accelerometer, angular velocity sensor and magnetic sensor. The information from these sensors allows us to calculate the motion locus. We focus here on data from a three-axis angular velocity sensor because although it is simple, the

Manuscript received August 15, 2017; revised October 20, 2017.

Toshiki Iso is with Research Laboratories, NTT DOCOMO, INC. Kanagawa, Japan (e-mail: isot@nttdocomo.com).

motion locus in the sensor coordinate system is sufficient to allow the detection of rotational motion. It is also difficult to calculate the motion locus in an inertial coordinate system with a high degree of accuracy.

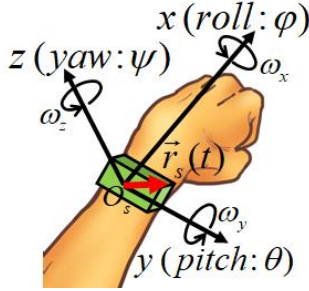


Fig. 3. Calculating motion locus.

To calculate the motion locus in the sensor coordinate system, we use roll angle φ , pitch angle θ and yaw angle ψ as orientation angles of the target object (user's wrist) on which the angular velocity sensor is mounted (see Fig. 3). By using quaternion q_1, q_2, q_3 and q_4 , the orientation angles (φ, θ, ψ) and the angular velocity $\Omega = (\omega_x, \omega_y, \omega_z)$ can be represented as follows [9];

$$\begin{pmatrix} q_1 \\ q_2 \\ q_3 \\ q_4 \end{pmatrix} = \begin{pmatrix} -\cos(\frac{\psi}{2})\cos(\frac{\theta}{2})\sin(\frac{\varphi}{2}) + \sin(\frac{\psi}{2})\cos(\frac{\theta}{2})\sin(\frac{\varphi}{2}) \\ -\cos(\frac{\psi}{2})\sin(\frac{\theta}{2})\cos(\frac{\varphi}{2}) - \sin(\frac{\psi}{2})\cos(\frac{\theta}{2})\sin(\frac{\varphi}{2}) \\ \cos(\frac{\psi}{2})\sin(\frac{\theta}{2})\sin(\frac{\varphi}{2}) - \sin(\frac{\psi}{2})\cos(\frac{\theta}{2})\cos(\frac{\varphi}{2}) \\ \cos(\frac{\psi}{2})\cos(\frac{\theta}{2})\cos(\frac{\varphi}{2}) + \sin(\frac{\psi}{2})\sin(\frac{\theta}{2})\sin(\frac{\varphi}{2}) \end{pmatrix} \quad (1)$$

Then,

$$\begin{pmatrix} \dot{q}_1(t) \\ \dot{q}_2(t) \\ \dot{q}_3(t) \\ \dot{q}_4(t) \end{pmatrix} = \frac{1}{2} \begin{pmatrix} -q_4(t) & -q_3(t) & q_2(t) \\ q_3(t) & -q_4(t) & q_1(t) \\ -q_2(t) & q_1(t) & -q_4(t) \\ q_1(t) & q_2(t) & q_3(t) \end{pmatrix} \begin{pmatrix} \omega_x(t) \\ \omega_y(t) \\ \omega_z(t) \end{pmatrix} \quad (2)$$

Rotation matrix $R(t)$ at time t is derived from the quaternion [10] by time integration of (2) and calculating the initial orientation angles based on (1).

$$R = \begin{pmatrix} 1-2(q_2^2+q_3^2) & 2(q_1q_2+q_3q_4) & 2(q_1q_3-q_2q_4) \\ 2(q_1q_2-q_3q_4) & 1-2(q_1^2+q_3^2) & 2(q_2q_3+q_1q_4) \\ 2(q_1q_3+q_2q_4) & 2(q_2q_3-q_1q_4) & 1-2(q_1^2+q_2^2) \end{pmatrix} \quad (3)$$

Thus, we can obtain $\vec{p}(t)$ at time t as the motion locus of the target object.

$$\vec{p}(t) = R(t)\vec{p}(t_0) \quad (4)$$

We define $\vec{p}(t_0)$ as the initial position.

$$\vec{p}(t_0) = \begin{pmatrix} r_0 \\ 0 \\ 0 \end{pmatrix} \quad (5)$$

B. Detecting Candidates of Normal Vectors

We have some choice in how to detect candidate's normal vectors from the motion locus. In this paper, we propose two approaches: cross product and plane approximation in local time. The former calculates cross product between motion locus and their time differential vector as the candidate's normal vectors. The latter calculates normal vectors of the plane equations that are approximately calculated from motion locuses in local time as the candidate's normal vectors. We will evaluate whether two methods can calculate the candidates normal vectors.

1) "Cross product" approach

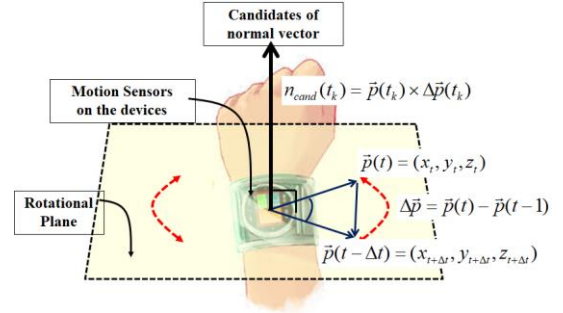


Fig. 4. Cross product approach.

We assume a plane that contains motion locus $\vec{p}(t_k)$ and time differential vector $\Delta\vec{p}(t_k)$ (see Fig. 4). Then, we calculate time differential vector $\Delta\vec{p}(t_k)$ from motion locus $\vec{p}(t_k)$.

$$\Delta\vec{p}(t_k) = \vec{p}(t_k) - \vec{p}(t_{k-1}) \quad (6)$$

This yields candidates of normal vectors $\vec{n}_{cand}(t_k)$ to the plane by the cross product approach as follows;

$$\vec{n}_{cand}(t_k) = \vec{p}(t_{k-1}) \times \Delta\vec{p}(t_k) \quad (7)$$

Then, $u \times v$ is the cross product of u and v .

2) "Plane approximation" approach

This subsection describes the other approach to detect normal vector candidates. We assume a plane that has almost holds $\vec{p}(t_k)$ at local time δ (see Fig. 5). In order to calculate a plane equation, we minimize the below cost function $L(\vec{n}(t_k), \Lambda)$ under the condition that normal vectors $\vec{n}$ are unit vectors.

$$L(\vec{n}(t_k), \Lambda) = \sum_{t_k \in \delta} (ux_{t_k} + vy_{t_k} + wz_{t_k} + 1)^2 + \Lambda\{1 - (u^2 + v^2 + w^2)\} \quad (8)$$

where $\vec{n} = (u, v, w)$. The conditions that minimize cost function $L(\vec{n}(t_k), \Lambda)$ are as follows

$$\frac{\partial L}{\partial u} = 0, \frac{\partial L}{\partial v} = 0, \frac{\partial L}{\partial w} = 0, \frac{\partial L}{\partial \Lambda} = 0 \quad (9)$$

By satisfying the above constraints, we can obtain the following singular value equation.

$$BX = \Lambda X \quad (10)$$

Matrix $B = B_{ij}$ in (if $i = 1; 2; 3; j = 1; 2; 3; 4$) is given by;

$$B_{ij} = \sum_{t_k \in \delta} p_i(t_k) p_j(t_k) \quad (i, j = x, y, z) \quad (11)$$

matrix B can be decomposed to left singular vector $\vec{U}$, the singular matrix S and the right singular vectors $\vec{V}$ as follows;

$$S = \vec{U} S \vec{V}^T \quad (12)$$

Minimum singular value $\Lambda_{\min Col}$ yields the minimum cost function $L(\vec{n}, \Lambda)$. Therefore, we can obtain the column vector $U_{3 \times \min Col}$ in the left singular vectors U as the normal vector candidates.

$$U_{3 \times \min Col} = \begin{pmatrix} u_{\min Col} \\ v_{\min Col} \\ w_{\min Col} \end{pmatrix} \quad (13)$$

We use the obtained normal vectors as candidates of normal vectors $\vec{n}_{cand}(t_k)$ on rotational planes.

$$\vec{n}_{cand}(t_k) = U_{3 \times \min Col} \quad (14)$$

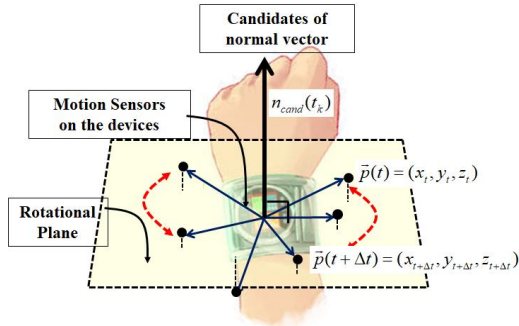


Fig. 5. Plane approximation approach.

C. Detecting the Representative Normal Vectors

Next, we calculate the deviations of normal vector candidates, $\vec{n}_{cand}(t_k)$ during local time by principle component analysis (PCA). PCA is used for the analysis of deviations in a distribution. Eigenvalues and eigenvectors stand for the deviations along the eigenvectors and axes of each distribution, respectively. Basically, PCA yields three eigenvalues and the distribution normally has a three-dimensional shape. If the minimum eigenvalue is nearly equal to zero, the distribution has two-dimensional shape, not three (see Fig. 6). That is to say, the motion locus is a two-dimensional figure and the eigenvector of the minimum eigenvalue can be a rotational axis. Therefore, by analyzing $\vec{n}_{cand}(t_k)$ with PCA, we can detect the representative normal vector in the observation time (See fig.7).

$$A(t_k)X = \lambda X \quad (15)$$

$\vec{N}(t_k)$ is the representative normal vector in local time ϵ with the minimum eigenvalue $\vec{e}(\min Dim)$ that solves the above eigen equation.

$$\vec{N}(t_k) = \vec{e}(\min Dim) \quad (16)$$

D. Calculation Criteria for Rotational Motions

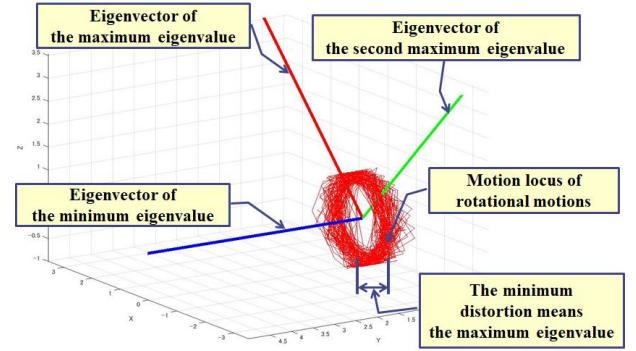


Fig. 6. Analysis of motion locus with PCA.

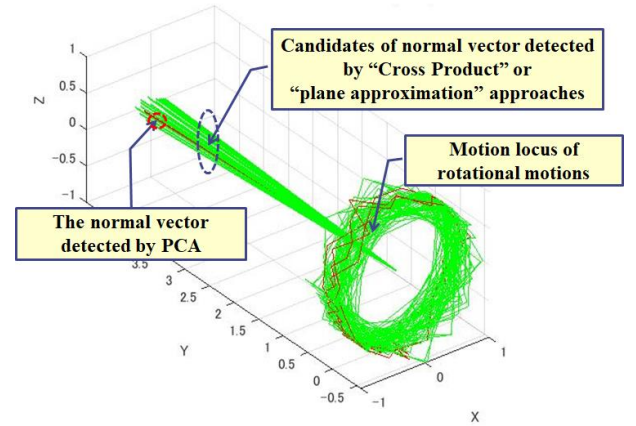


Fig. 7. Extracting the representative normal vector from candidates of normal vector.

As described above, the motion locus becomes flatter as the normal vector deviation falls. Therefore, we define criteria $CFR(t_k)$ for rotational motion based on the results of PCA.

$$CFR(t_k) = \log_{10} \{RMEV(t_k)\} \quad (17)$$

where

$$RMEV(t_k) = \frac{\lambda(\min Dim)}{\sum_{dim} \lambda(dim)} \quad (18)$$

If a target object undergoes a rotation motion, the locus's distribution becomes flat. Therefore, the minimum eigenvalue is smaller than the other eigenvalues, that is to say, $CFR(t_k)$ is becoming small.

III. EVALUATION

We verify that the proposed method can detect rotational motion. We do so by using simulation data generated by adding random noise to an ideal rotational motion locus. The ideal rotational motion locus forms perfectly flat figures on the x-z plane. Fig. 8, 9, 10, 11 show the results of our proposed cross product and plane approximation methods,

respectively. The upper and the lower graphs in the figures show random noise and criteria for rotational motion $CFR(t_k)$ over time, respectively. When the motion locus is noise-free, $CFR(t_k)$ is nearly zero, the other side, $CFR(t_k)$ is not stable because of random noise. Thus the cross product

approach is only slightly more sensitive to noise than the plane approximation approach. Both methods could detect almost all changes in rotational motion.

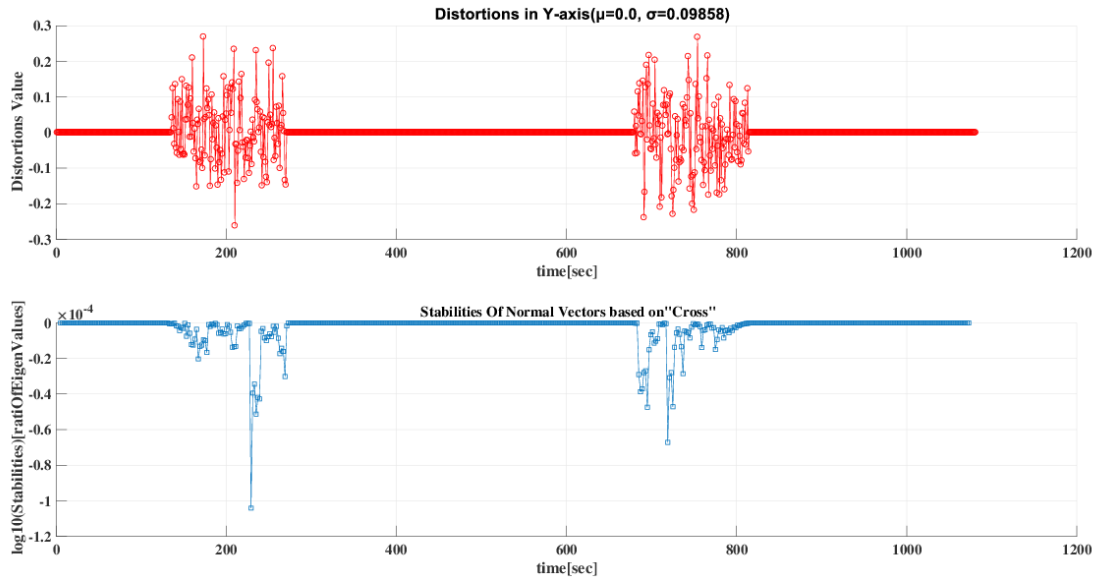


Fig. 8. Detecting rotational motions by “cross product” approach (pattern 1).

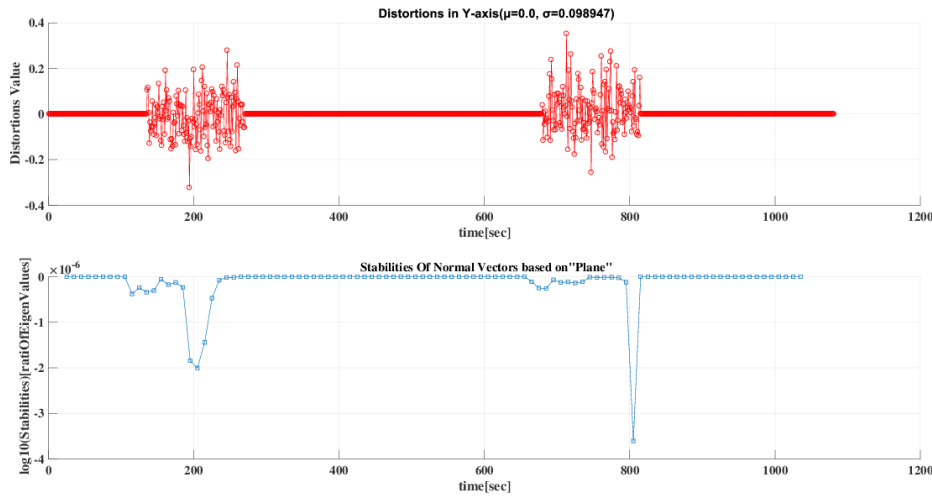


Fig. 9. Detecting rotational motions by “plane approximation” approach (pattern 1).

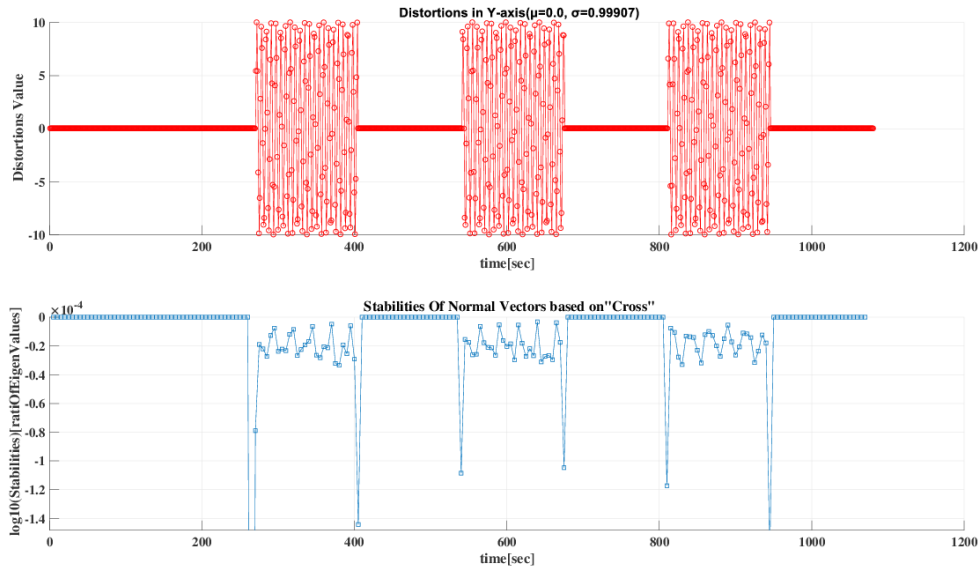


Fig. 10. Detecting rotational motions by “cross product” approach (pattern 2).

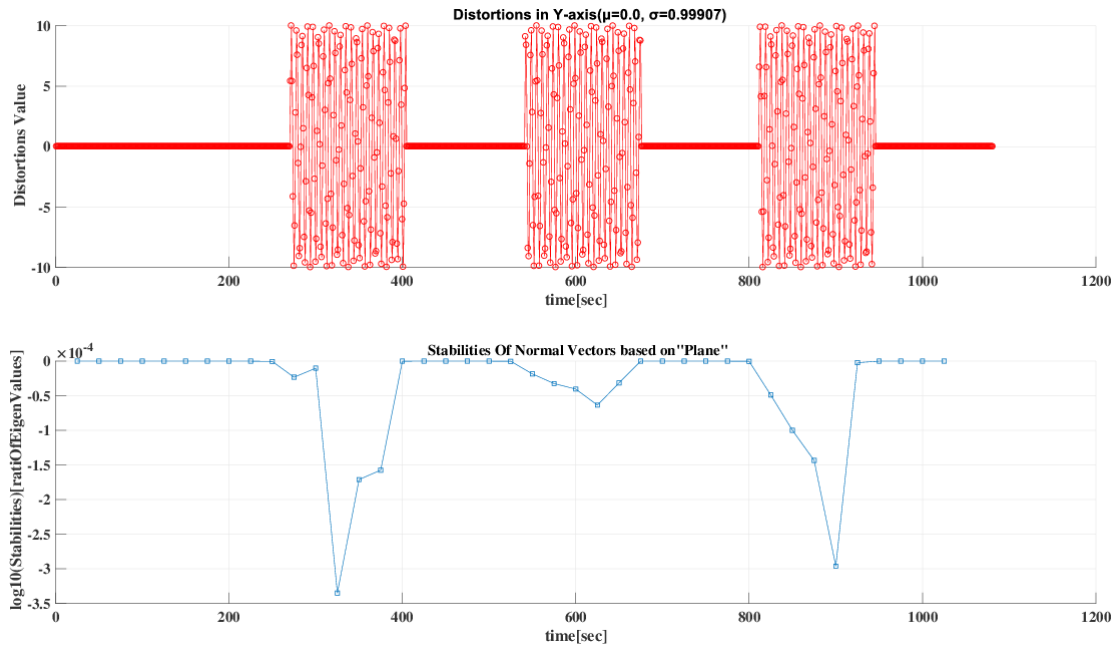


Fig. 11. Detecting rotational motions by "plane approximation" approach (pattern 2).

IV. CONCLUSION

We proposed criteria for detecting rotational motion from motion locus. Simulations showed that our proposed methods of cross product and plane approximation, which use PCA, were able to detect rotational motions. As our proposal needs only a gyro-sensor and has low computation cost, it is easy to implement in mobile devices for mobile services such as command interfaces and canine healthcare. In future works, we will evaluate the method by processing sensory data actually captured by wrist-watch type and dog collar type devices.

ACKNOWLEDGMENT

The author would like to thank Dr. S. Hiyama, NTT DoCoMo Research Labs for his encouragement.

REFERENCES

- [1] NTT DOCOMO. Moveband. [Online]. Available: <http://www.d-healthcare.co.jp/products/moveband3/about/>
- [2] NTT DOCOMO. Petfit. [Online]. Available: <https://www.docomopet.com/>
- [3] S. A. Sajjadi, D. Moazen, and A. Nahapetian, "Airdraw: Leveraging smart watch motion sensors for mobile human computer interactions," arXiv preprint arXiv:1705.02689, 2017.
- [4] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga, "Complex human activity recognition using smartphone and wrist-worn motion sensors," *Sensors*, vol. 16, no. 4, p. 426, 2016.
- [5] C. Xu, P. H. Pathak, and P. Mohapatra, "Finger-writing with smartwatch: A case for finger and hand gesture recognition using smartwatch," in *Proc. the 16th International Workshop on Mobile Computing Systems and Applications*, ACM, 2015, pp. 9–14.
- [6] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, p. 33, 2014.
- [7] J. Ruiz, Y. Li, and E. Lank, "User-defined motion gestures for mobile interaction," in *Proc. the SIGCHI Conference on Human Factors in Computing Systems*, ACM, 2011, pp. 197–206.
- [8] T. Iso and K. Yamazaki, "Gait analyzer based on a cell phone with a single three-axis," *ACM MobileHCI '06*, pp. 141–144, 2006.
- [9] J. M. Cooke, M. J. Zyda, D. R. Pratt, and R. B. McGhee, "Npsnet: Flight simulation dynamic modeling using quaternions," *Presence: Teleoperators & Virtual Environments*, vol. 1, no. 4, pp. 404–420, 1992.
- [10] A. M. Sabatini, "Quaternion-based extended kalman filter for determining orientation by inertial and magnetic sensing," *IEEE Transactions on Biomedical Engineering*, vol. 53, no. 7, pp. 1346–1356, 2006.



Toshiki Iso received the Ph.D. Degree in engineering from Waseda University, Tokyo, Japan, in 2008. Since joining NTT Human Interface Laboratories in 1988, he has been studying image processing and pattern recognition for mobile phone. He is currently working for Research Laboratories, NTT DOCOMO. He is a member of IEICE. and IPSJ.