State Localization of EVA Service Robot

Hiroki Kato, Hirotaka Sawada, Kazuya Konoue & Mitsushige Oda

Institute of Aerospace Technology, Japan Aerospace Exploration Agency (JAXA)

Abstract
JAXA envisions that next-generation space robotics technologies are developed on JEM (Japanese Experiment Module at International Space Station). The autonomous EVA service robot has capabilities of construction, inspection, and communication with astronauts in future space missions, and is scheduled to be sent there in 2011. The robot is intended to help astronaut's EVA (Extra-Vehicular Activities), or even replace for them when assigned tasks are simple and tedious.

As high-level of precision of control is required, it is critical to accurately compute a position of the robot. This paper states an overview of the EVA Service Robot, and the algorithm "state localization," identifying an accurate position of the robot

EVA支援代行ロボットの同定アルゴリズム・State Localization: JAXA宇宙ロボティックスの発展戦略として、将来の有人宇宙活動に必要となるキー技術を国際宇宙ステーション(ISS)日本実験棟(JEM)を利用して確立することが掲げられている。そういう経緯からJEM船外プラットフォームを利用して建設能力、検査能力、管制能力を実証する「自律EVA支援・代行ロボット」の開発を2011年打上げ予定に向けて進めている。
　限られたリソースの中で高い安全性が求められる本実験ではロボットが自己の位置を正確に知ることがクリティカルとなる。本論文ではその自己位置決定に使用される同定アルゴリズム「State Localization」を示し、EVA 支援・代行ロボットに適応してみる。

1. Background

Localization of Space Robotics

People like safety assessment reviewers would comment that autonomous robots are generally dangerous. Those people would be uncomfortable when autonomous robots take actions at places where they cannot watch because the robot may collide with a critical system and destroy it. When robot localizes her position accordingly, the robot can counteract for safety. Thus, we must establish a baseline of robot's localization method.

The EVA Service Robot

Autonomous robotics is required when (1) robot(s) work at a place where human cannot communicate instantaneously, and (2) an amount of tasks is overwhelming for people (astronauts or ground operators). For (1), Hayabusa navigated autonomously around/on Itokawa, where the communication between the expletory vehicle and ground operators took 33 minutes. For (2), although we expect its necessity for future missions like building a moon base or Solar Space Power System (SSPS), we have not shown its usefulness.

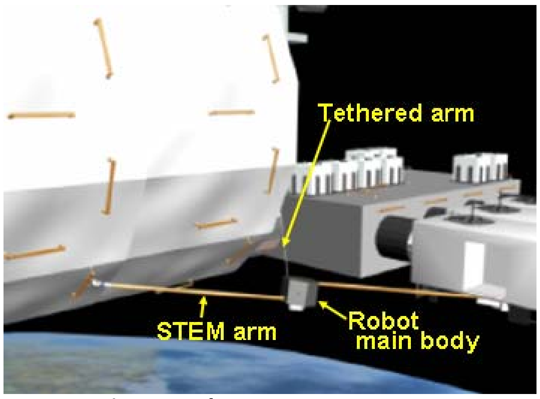


Figure1. EVA service robot in REX-J

The EVA service robot (Figure1) is experimented on the JEM's exposed facility, and the experiment is called REX-J (Robot Experiment on JEM). REX-J is a candidate of the second-term experiment in JEM's exposed facility, which is scheduled to launch in 2011. The EVA Service Robot is intended to take a major role of future space missions. With a great moving capability, she will be able to communicate with astronauts, construct space building structures, and inspect the places where the astronauts can hardly reach. This robot is intended to use for future space missions, such as constructions and maintenance of a moon base and Space Solar Power System. In Table1, the EVA service robot is compared with the CANADARM2, the most famous existing space robot for a construction purpose.

Robot	CANADARM2	EVA Service Robot
Handling Target	Trusses for construction and space shuttle	Light-weight (like inflatable) structures
Weight	4000kg	50kg+
Power	4.8kW (peak)	40W+ (peak)
Size	17.6m arm	500x500x300mm+
Mobility	Rolling on the Mobile Transporter	Tethered arms

Table1. CANADARM2 vs. EVA Service Robot

Configuration in REX-J

The JEM's exposed facility will provide resources to the robot, and they are limited as follows:

- *Total Size: 500x500x300mm*
- *Total Weight: 50kg*
- *Total Electric Power: 40W*
- *Communication Bandwidth: 5kbps*

With this little amount of electric power, a robot cannot perform heavy-duty work at all. We concluded that it is unrealistic for the EVA service to experiment construction capability in REX-J. This time, we focus on demonstrating robot's moving capability. The robot has a high definition camera, and can inspect the outer wall of the JEM facility. Two-millimeter-diameter holes created by corrosions of space debris are detected, and the clear images are provided to the user of the robot.

Figure2 is the block diagram of the EVA service robot in the REX-J configuration. In REX-J, since only the JEM platform can provide electric power directly to the EVA service robot, she cannot leave from the dock. The electricity and communication link is wired through the STEM "leg" and forwarded electricity to the main body of the robot.

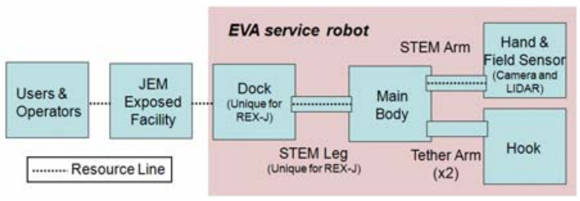


Figure2. Block diagram of REX-J

2. Functionalities

Moving around space:

While the mobile transporter is placed for the CANADARM2 to move around the ISS, the EVA service robot moves by holding handrails used for astronauts' EVA, just like astronauts do. Since the handrails exist throughout the ISS, the robot can move there, just like an astronaut can do.

The EVA service robot has two kinds of arms: STEM arms and tethered arms. Each arm is expected to stretch up to about 10 meters. The tethered arms are not rigid and are passive so they cannot move by themselves. The STEM arm is rigid and can hold the tethered arm. By controlling the direction and length of the STEM arm, the tethered arm can be attached to / removed from one of the handrails on the ISS. The dock has two degree of freedom in azimuth and elevation to aim the moving direction, and the main body of the robot can rotate in her yaw axis. Then, by controlling the length and tension of the tethered arms, the main body of the robots can move around the space that is inside the anchor points by the arms. When it has three anchor points, it can move a plane. Also, when it has four anchor points, it can move a three-dimensional space (Figure3). When the EVA service robot wants to move to the area outside a movable region, the STEM arm can remove the tethered arm at the handrail that is not related to the new region, and place at a next handrail that enables the robot to reach to the aimed region.

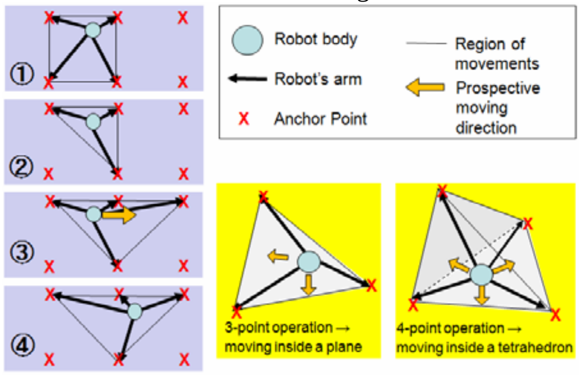

①
②
③
④
Robot body
Robot's arm
Anchor Point
Region of movements
Prospective moving direction
3-point operation → moving inside a plane
4-point operation → moving inside a tetrahedron

Figure3. EVA Service Robot moving principal

Grabbing the handrail:

The major challenge of the REX-J is to grab the handrail. To achieve this, the STEM arm shall extend 10m. A STEM (Figure4) is a proven technology that was used for the Hubble Telescope's repeated extension capability [1]. There is a dram to stretch/shrink the arm in the side attached to the robot's main body. In the other end of the stem, a robotic hand is attached. Let us call it "the STEM-hand". The hand has three-degree of freedom that makes the robot possible to approach the handrail. On the STEM's hand, there is a camera (and/or LIDAR) that senses the environment and identify the target as seen in the block diagram of REX-J.

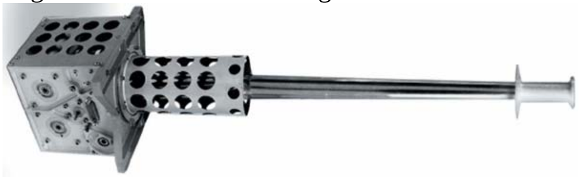


Figure4. A STEM

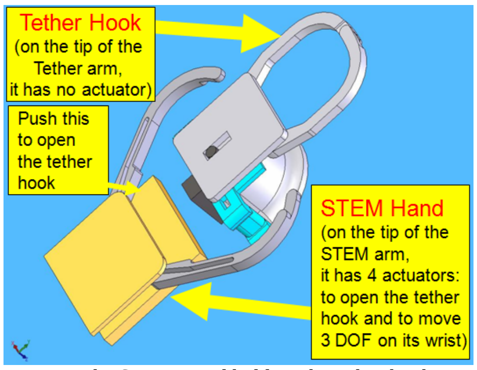
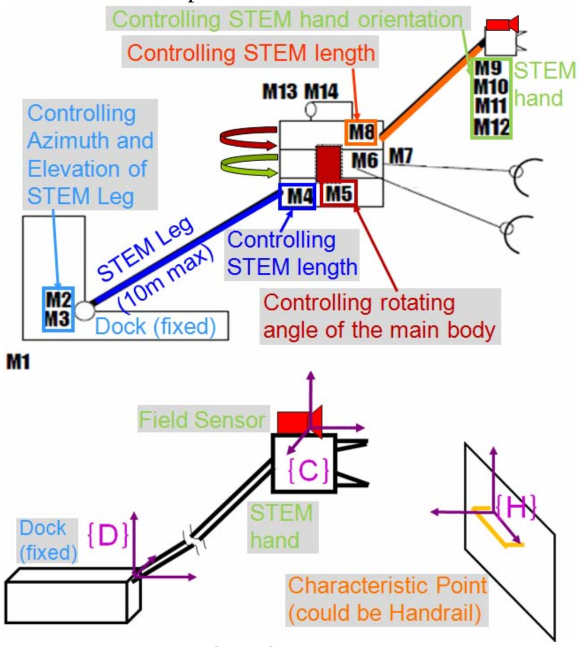

Tether Hook
(on the tip of the Tether arm, it has no actuator)
Push this to open the tether hook
STEM Hand
(on the tip of the STEM arm, it has 4 actuators: to open the tether hook and to move 3 DOF on its wrist)

Figure5. The STEM Hand holding the tether-hook

The STEM hand has a capability to hold the tip of the tethered arm (Figure5). The hook is attached at the end of the tethered arm. Let us call it the tether-hook. The hand on the STEM arm can grab it, and open/close the hook safely. At the opposite end of the tethered arm (that is on the main body), there is a mechanism that can adjust the length and tension of the tether.

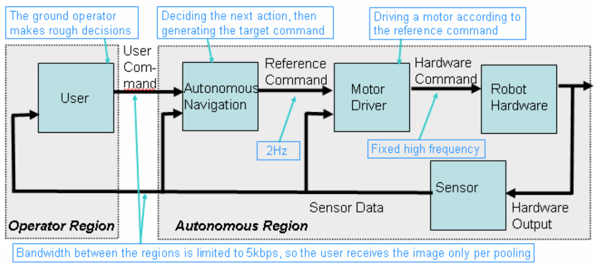
Figure6 shows the coordinate system of the robot with 14 motors. The depth of the handrail is only 6cm, so the robot shall locate her position with accuracy of about 2cm at worst. When the STEM arms are stretched, the total length of the arms can reach 20m, and the error becomes large in this case. For example, if the robot dock's motor angle has one degree of error, the position of hand-camera at the tip of the STEM hand has 35cm of error by solving kinematics, which is not acceptable.


Controlling STEM hand orientation
Controlling STEM length
M13 M14
M9
M10
M11
M12
STEM hand
Controlling Azimuth and Elevation of STEM Leg
M8
M6
M7
M4
M5
STEM Leg (10m max)
Controlling STEM length
M2
M3
Dock (fixed)
Controlling rotating angle of the main body
M1
Field Sensor
{C}
Dock (fixed)
{D}
STEM hand
{H}
Characteristic Point (could be Handrail)
Figure6. 14 actuators and coordinate system

Autonomous Operation:

Figure7 shows the control block diagram of the system. The outer control loop includes heuristic inputs of the ground operators, while the rest of loops are autonomous. When the outer control loop is frequent, the ground operators should be able to tune the position parameters by watching the images of the hand-camera. However, in the REX-J, since the communication bandwidth is limited to 5kbps, it takes many seconds to download the image. Thus, the robot must autonomously close the control loop for the most of time, and the ground operators can only confirm the robots' operations at critical occasions intermittently. The autonomous navigation block in Figure7 takes a user command, receives current sensor data from hardware, and forwards a reference command to robot's hardware.

Figure8 is the inside the autonomous navigation block in Figure7. *The Mode Arbiter* decides the mode of an operation depending on the state of the robot and user command. *The Trajectory Generator* generates a path of the robot depending on a user command and feedback data. *The Tracker* calculates target hardware parameters based on the trajectory. Given an image of the camera, *the Image Analyzer* finds characteristic points to calculate the current position (that is to find $\{T_C{}^H\}$). *The State Localizer* finds the current state of the robot (position and velocity) based on sensor data and *the Image Analyzer*, and refines the parameters of position and camera.


The ground operator makes rough decisions
Deciding the next action, then generating the target command
Driving a motor according to the reference command
User
User Com-mand
Autonomous Navigation
Reference Command
Motor Driver
Hardware Command
Robot Hardware
2Hz
Fixed high frequency
Sensor
Sensor Data
Hardware Output
Operator Region
Autonomous Region
Bandwidth between the regions is limited to 5kbps, so the user receives the image only per pooling
Figure7. Control diagram

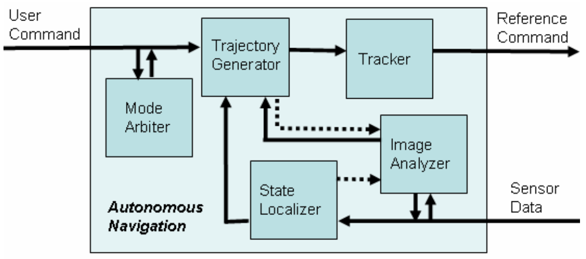


Figure8. Inside Autonomous Navigation

Locating a current position of the robot:

In order to operate an autonomous robot, she must locate her current state accurately. Especially, to grab the handrail, the robot must accurately evaluate $\{T_D{}^H\}$ (defined as the transformation from the robot dock's $\{D\}$ coordinate system to the target handrail's $\{H\}$ coordinate system). The sensors on the STEMs return linear positions, and the other rotating motor sensors return angular data. If there is no position error at all, $\{T_D{}^C\}$ is found perfectly. $\{C\}$ is the coordinate system of the field sensor, which is mounted on the hand rigidly. Evaluating $\{T_C{}^H\}$ using a computer vision algorithm without error, $\{T_D{}^H\}$ is found perfectly. However, in practical, there is error in $\{T_D{}^C\}$ as it would be difficult to home the zero position in space. Let us call the algorithm of localizing robot's location (finding $\{T_D{}^H\}$ for now) "state localization," the topic of this paper.

3. Algorithm

When robot's operation is not for an initial exploration purpose, a global map with some degree of details should be given. The operators should be able to preplan the robot's trajectory, which is also true for REX-J. The given map should be accurate, but, in practical, it is not always true because of two reasons: (a) measurement error during initial global map creation, and (b) change of the map (environment) after map creation. For REX-J, there is no guarantee that (a) does not happen, and (b) may well happen because of ISS/JEM's physical change due to temperature change or other space environmental factors. If the robot only trusts the pre-computed map, the robot may crush on an obstacle that is outside robot's trajectory or that is placed after creation of the map. Therefore, the robot must localize her position (adjusting and updating the map according to robot's sensor data), which is well-proven technology [2].

To update the preplanned map, robot's internal parameters must be very accurate; otherwise, the updated map will also be erroneous. The robot's internal parameters are transformations of the links and sensors including the motor angles, link length, distance between sensors in sensor fusion, and camera intrinsic parameters. Although we calibrate sensors and their positions as needed prior to the operation for any robotics applications, their accuracy may be ruined any time during/before the operation, (physical shock during shipment of the robot, for instance.) For example, in REX-J the robot will encounter the shock due to rocket's liftoff, which is expected to be 30G, in spite of rigorous calibration on ground. Even for ground robots, changes of internal parameters during the operations or transportation, cause serious problems. Thus, it is important for robots to equip capability to calibrate internal parameters dynamically during the operation.

In our state localization algorithm (Figure9), let us apply Levenberg-Marquardt Algorithm (LMA), a good method for solving a non-linear least-square minimization problem [3][4] to adjust the robot's internal parameters and then the global (external) map. The robot's internal system includes motor angles and length between links, to obtain accurate internal parameters. After settling a model function vector "r" to minimize and n-by-1 variable vector "x" (usually represents error quantities), the following equation is solved for "p_k" iteratively (k = 1,2,…):

$$(J_k^T J_k + \lambda_k I)p_k = -J_k^T r_k$$

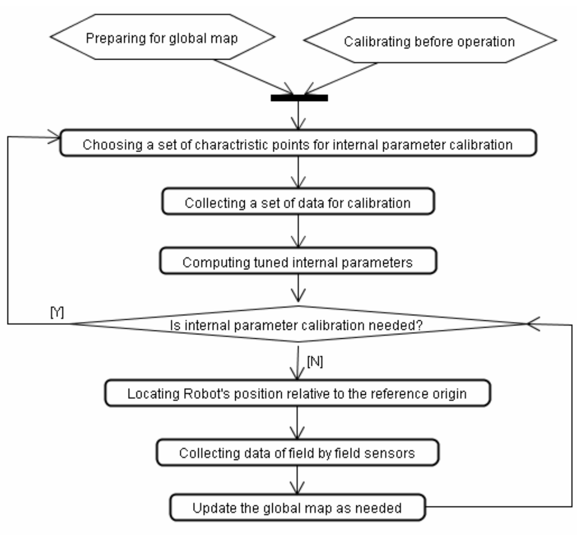
where

$$r_k(x) = \begin{bmatrix} r_1(x_k) \\ \vdots \\ r_m(x_k) \end{bmatrix}$$

$$J_k(x) = \left[\frac{\partial r_j}{\partial x_i}\right]_{\substack{j=1,\ldots,m \\ i=i,\ldots,n}} = \begin{bmatrix} \nabla r_1(x_k)^T \\ \vdots \\ \nabla r_m(x_k)^T \end{bmatrix}$$

$$p_k = x_{k+1} - x_k$$

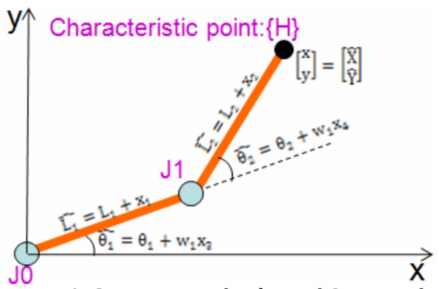
and, lamda is an adjusting constant to scale J'J less than a desired radius. The idea is that, while famous Tsai's Algorithm (commonly employed for tuning camera's intrinsic parameters [5]) uses LMA by modeling camera's intrinsic parameters, our state localization algorithm apply LMA for the rest of the internal system, including motor angles and length between links, to obtain accurate internal parameters before start tuning the global (external) map. Detailed usage of the algorithm for the EVA Service Robot is described in the next chapter, as it should be able to widely apply for other applications including ones on ground.


Preparing for global map
Calibrating before operation
Choosing a set of charactristic points for internal parameter calibration
Collecting a set of data for calibration
Computing tuned internal parameters
[Y]
Is internal parameter calibration needed?
[N]
Locating Robot's position relative to the reference origin
Collecting data of field by field sensors
Update the global map as needed
Figure9. Overview of State Localization Algorithm

4. Application

Calibrating Internal Parameter of the robot:

In practical, the CPU of the robot has to be reset in space applications for recovery when the memory data is ruined (called single events). Whenever they happen, the motor's absolute angle information is reset. The motor physically pulses at certain point (say, at zero degree or at physical limitation angles) and reset the counter. Also, an absolute encoder will be installed so that the motor does not lose its position information when the system is restarted. Even though this method should solve most situations, the robot cannot perform the reset method in certain situations, where we cannot confirm clearance of the robot's movements. For such a situation, we can apply the state localization algorithm to calibrate the internal parameters. Also, the STEM arm stretches, it is expected that the arm twists in its yaw direction, which cannot be measured by any sensor explicitly. After we model the phenomena, we can find the parameters in the model by using the state localization algorithm.



y
Characteristic point:{H}
$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \hat{X} \\ \hat{Y} \end{bmatrix}$
$\widehat{L_2} = L_2 + x_2$
$\widehat{\theta_2} = \theta_2 + w_1 x_4$
J1
$\widehat{L_1} = L_1 + x_1$
$\widehat{\theta_1} = \theta_1 + w_1 x_3$
J0
X

Figure10. Generic Example of Partial State Localization

Let us discuss the most generic situation seen in Figure10, which would be expected during REX-J's operation. I.e., Letting the joint at the origin J_0 be M_3 (elevation of the dock) or M_{10} (pitch angle of the STEM hand) and J_1 be the field sensor on the tip of the STEM hand {C} and fix the rest of the motor angles, M_3, M_{10}, and the STEM length can be calibrated.

In theory, we could calibrate any combination of the parameters at a time, but by picking few (at least 1) on the same plane, we could setup the problem more easily and accurately. In this situation in the figure, we could tune at least four different parameters by applying the same set of the simple equations:

$$\begin{cases} r_{2k-1}(x) = \widehat{L_1}\cos(\widehat{\theta_1}) + (\widehat{L_1} + \widehat{L_2})\cos(\widehat{\theta_1} + \widehat{\theta_2}) - \hat{X} \\ r_{2k}(x) = \widehat{L_1}\sin(\widehat{\theta_1}) + (\widehat{L_1} + \widehat{L_2})\sin(\widehat{\theta_1} + \widehat{\theta_2}) - \hat{Y} \\ x_0 = [0 \quad 0 \quad 0 \quad 0]^T \\ \widehat{L_1} = L_1 + x_1, \quad \widehat{L_2} = L_2 + x_2, \\ \widehat{\theta_1} = \theta_1 + w_1 x_3 \quad \widehat{\theta_2} = \theta_2 + w_1 x_4 \end{cases}$$

where w_1 is a weight constant converting from the length of link (probably in meter) and angle (probably in radians). When we compute the error vector step "p", we simply take a previously chosen length of the vector "x". Thus, although the weight constant is not necessarily required, it would give a good sense of the remaining error between different quantities (length and angle). The robot collects "k" sets of data points (L, theta, X-hat and Y-hat) by moving appropriate links, and obtains the error vector "x". In the equation above, we assume a constant error throughout the system, and we can apply the same technique for any kind of error as long as the model functions "r" are differentiable. Also, in the equation above, the target characteristics points, X-hat and Y-hat, (which are transformed from the 3d-representation in the global map,) are assumed to be very accurate in the given global map. In this sense, the physical setup could be difficult.

While we obtained only 2D position information in the example above, we could use 6DOF information as needed. By setting up appropriate model functions for each error, the robot can obtain accurate internal parameters.

Modifying the Global Map of the robot:

When the robot knows that the internal parameters are accurate, the robot should trust the field sensor data more than the pre-computed global map. As the robot obtains field sensor data, she can create point cloud and then can conduct obstacle detection routine and generate trajectory. Also, a real-time global map is obtained by correlating characteristic segments of sensor data with the pre-computed global map.

5. Summary and Next Steps

We have presented the state localization algorithm for polishing robot's current position in a given global map. Since the algorithm assumes well-calibrated internal parameters, process of calibrating them is also presented as a part of the routine. The state localization algorithm has a room for improvements.

A. Tuning map and internal parameters together dynamically. Although the setup equation for tuning the internal parameters requires accurate target characteristics point positions (X-hat and Y-hat) , it would not be practical to find them everywhere on the map. The algorithm would be ready for practical use if it can accept little position error of characteristic points, namely:

$$\begin{cases} r_{2k-1}(x) = \widehat{L_1}\cos(\widehat{\theta_1}) + (\widehat{L_1} + \widehat{L_2})\cos(\widehat{\theta_1} + \widehat{\theta_2}) - \hat{X} \\ r_{2k}(x) = \widehat{L_1}\sin(\widehat{\theta_1}) + (\widehat{L_1} + \widehat{L_2})\sin(\widehat{\theta_1} + \widehat{\theta_2}) - \hat{Y} \\ x_0 = [0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0]^T \\ \widehat{L_1} = L_1 + x_1, \quad \widehat{L_2} = L_2 + x_2, \\ \widehat{\theta_1} = \theta_1 + w_1 x_3 \quad \widehat{\theta_2} = \theta_2 + w_1 x_4 \\ \underline{\hat{X} = X + w_2 x_5} \quad \underline{\hat{Y} = Y + w_2 x_6} \end{cases}$$

where X and Y represent a point on the global map *that can contain some degree of error*, and w_2 is a weight constant like w_1. Those equations model both internal and external error; however, according to the result of the simulation, the error vector "x" converges to a point giving a minimum error, (which is mathematically correct,) but did not correct error accurately. This is probably because there are several points satisfying the minimum condition. To overcome the situation, we may have to add more constraints or take a different approach.

B. Modeling the error. In the equations above, we assume constant errors. Especially, the STEM arm is under development and we have not properly modeled the physical system yet. Also, we added physical transformation of the field sensors (camera and LIDER) into the internal parameters (L_2 and $theta_2$). Modeling of those parameters has to be more careful.

References

[1] F.P.J. Rimrott, G. Fritzsche, Fundamentals of STEM Mechanics, IUTAM-IASS Symposium on Deployable Structures: Theory and Applications, (2000), pp. 321-333

[2] C. Urmson, W.L. Whittaker, et. al, A Robust Approach to High-Speed Navigation for Unrehearsed Desert Terrain, Journal of Field Robotics, Vol. 23 (2006) pp. 467-508

[3] D.W. Marquardt, An algorithm for least squares estimations estimation of non-linear parameters, SIAM Journal, 11 (1963), pp. 431-441

[4] J. Nocedal, S. J. Wright, Numerical Optimization 2nd edition, Springer Science + Business Media, (2006), pp. 245-269

[5] R. Tsai, A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV camera, IEEE Journal of Robotics and Automation RA-3 (4) 323-344