Mobile Robot Pose Estimation Based on Position/Velocity Sensor Fusion

Kirill Krinkin, Artyom Filatov
Saint-Petersburg Electrotechnical University "LETI"
St. Petersburg, Russia
kirill.krinkin@fruct.org, art32fil@gmail.com

***Abstract*—An autonomous self driving platform receives information about environment using only its onboard sensors. And it seems obvious that using several sensors could provide more certain information with reduced measurement error [1]. But a general question is how to fuse measurements from different kinds of sensors (like a camera and an accelerometer) to get refined data about a platform or world state. This paper presents a theory based on groups that proves a possibility of correctness of error extraction from a moving model. And there are results of application this theory on fusing measurements from two sensors: odometer and scan matcher.**

I. Introduction

A task of sensor fusion is to refine a state of an object using observations from several different sensors. A human uses five senses and fuse them to get his best estimation of a world state. That means that a mobile platform could refine the world state too using several sensors, but it should rely on a determined algorithm. And one of the main faced issue is to find a way how to reduce errors and provide more accurate estimation about a world state from mobile autonomous self driving unit via sensor fusion.

There exist solutions and algorithms that illustrates how to reduce measurement error using several observations of one characteristic [2], for example distance or volume. It could use an average value or a weighted average. But the most challenging task is to combine data from pretty different sensors. Often this task is solved individually and there is no global idea for every sensor combination. For example very popular sensor combination is an accelerometer and a gyroscope. First sensor provides measures of current unit acceleration and second provides current changes of orientation angles in space. This fusion does not clarify measurements of sensors with other sensor data but provides more information than using only one of them. Another example of sensor fusion is a combination of data from a camera and an IMU sensor. This is a popular sensor set for autonomous driving or flying units and there exist solutions [3],[4], how to fuse these sensors to reduce measurement errors of both sensors. But all considered approaches of sensor fusion depends on exact type of sensors and it is a very challenging or impossible task to expand this approaches to every sensor set.

One way of refining sensors data in general is Kalman filter [5]. This filter tries to fuse data from sensors with prior knowledge about dynamic of process that is called model. There are some requirements to a model and observations in Kalman filter. For example it is claimed that the model and the observations have added linear error of every variable from process state. So this error could be easily extracted from equations. There is an extension to Kalman filter that is named EKF [6] that could handle unlinear model by expanding it to Taylor series. But this approximation has increasing error because of dropping a reminder. Moreover Kalman filter increases its complexity in dependence of a variable amount like $O(n^3)$, so if the amount of variables is increased during an observation process, Kalman filter becomes more complex much faster.

Kalman filter equations start from a minimization of an error functional that presents a sum of square norms of a model error and an observations error. This idea could be applied for any base of sensor fusion. The main issues of this idea could be presented with a following list:

1) Find out formulas describing observations of the selected sensors;
2) Mathematically extract error values for every observed variable;
3) Clarify formulas of error norms;
4) Find the minimum of a sum of square norms of errors.

The challenging problem is defined in item 2 of the previous list. It appears when observation error is not added and is involved in more complex way. In this case the first question is "is it possible to extract this error without loosing a correctness". The answer is that it is possible if there exist a group with some operator $\oplus$ and it is possible to present the error with this operator [7]. In other words the error could be extracted without loosing a correctness if the error is presented in an equation via $\oplus$ function that is an operator of a group and this group consists of elements with the same type as considered error (i.e. if an error is from $M_{n\times m}$, a group should have elements from $M_{n\times m}$).

In this paper two types of measurements are considered. First one is global measurements – regular measurements that provide a full state of observed characteristic and don't rely on any previous states. An example of there measurements is 3D coordinates of a mobile platform observed with GPS, landmark markers, localization algorithm etc. Second type of measurements is iterative measurements – that kind where every new observation relies on a previous one, so without any refining total error increases. An example is 3D coordinates observed with odometry, i.e. a movement from a previous observation. In general case the second type of sensors measures characteristics that are derivatives from observations measured by the first sensor type. In this case during a transformation from differential equations to linear difference equations there appears a dependency from previous states.

The rest of the paper is organized as follows: Section II gives a description of used mathematical algorithms and models. A description of considered environment is considered in Section III. A sensor fusion algorithm is described in Section IV. The application of math is presented in Section V. Finally, Section VI concludes the paper.

II. Mathematical preliminaries

This section presents considered mathematical models with explanations and some proves. There is a model of iterative odometry measurements, explanations of global measurements and group theory that proves a correctness of used models.

A. Groups theory

There is a definition of a group and some properties that could be useful below.

A group is a set G equipped with a binary operation $*$ that satisfy following rules [8]:

- closure: $\forall g_1, g_2 \in G : g_1 * g_2 \in G$;
- associativity: $\forall g_1, g_2, g_3 \in G : (g_1 * g_2) * g_3 = g_1 * (g_2 * g_3)$;
- identity: $\forall g \in G, \exists! e \in G : g * e = e * g = g$;
- invertibility: $\forall g \in G, \exists! g^{-1} \in G : g * g^{-1} = g^{-1} * g = e$.

An example of a group is square matrices with nonzero determinant and regular matrix multiplication as a group operation. This set and operation satisfy the axioms of group because multiplication of two square matrices is a square matrix, the identity matrix is neutral by multiplication, for every square matrix with nonzero determinant there exists an invert matrix and matrix multiplication is associative.

A group that is considered in this paper presents 3D vectors that consist of "x" and "y" coordinates on a plate and "θ" orientation angle on XOY plate [9]:

$$X = \begin{pmatrix} x & y & \theta \end{pmatrix}^T \in \mathbb{R}^3. \tag{1}$$

A group operation $\oplus$ between elements is "an addition in a direction" mathematically could be described in the following way:

$$X_1 \oplus X_2 = \begin{pmatrix} x_1 \\ y_1 \\ \theta_1 \end{pmatrix} \oplus \begin{pmatrix} x_2 \\ y_2 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} x_1 \\ y_1 \end{pmatrix} + R(\theta_1) \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \\ \theta_1 + \theta_2 \end{pmatrix}. \tag{2}$$

where $R(\theta)$ – a anticlockwise rotation matrix of an angle θ.

Geometrically this operation could be interpreted as usual addition of 2D coordinates but with changed angle of coordinate basis vectors.

Note that this operation is not commutative and in general: $X_1 \oplus X_2 \neq X_2 \oplus X_1$.

It is possible to check that chosen operation and set of vectors generate a group. So there are important group specific properties:

- Operation (2) is bijective;
- For every element X from (1) there exists X^{-1}.

B. Process dynamic

As it was mentioned above in Section I we consider two types of measurements that are going to be fused. First of them are direct "global" measurements. This observations are got independently from time. It means that observations collected at a moment t_i are independent from observations collected at a moment t_j. So the general equation of this kind of observations could have following structure:

$$Z_i = H_i X_i + \Gamma_i \zeta_i \tag{3}$$

where Z_i – vector of values – global observations of a state, X_i – a measured state, H_i – a matrix of observation that describes correspondence between the state and observations, ζ_i – error of measurements, Γ_i – a matrix of error that describes correspondence between the state and measurement error.

We have an assumption that observations could be presented in (3). In general influence of a state X_i and an error ζ_i are not linear, but we consider only this linear case. In other case this model could be linearized by Taylor series.

Using formula (3) it is possible to extract an error value ζ_i:

$$\zeta_i = inv(\Gamma_i) \left(Z_i - H_i X_i \right) \tag{4}$$

where $inv(\Gamma_i) = \Gamma_i^{-1}$ if Γ_i is a square matrix or $inv(\Gamma_i) = \left(\Gamma_i^T \Gamma_i\right)^{-1} \Gamma_i^T$ – a pseudo-invariant matrix in another case.

Now it is possible to use this formula for minimization the error ζ_i and find an estimation of X_i that provides minimum norm of the error.

Second type of measurements – iterative measurements – a case when a new observation does not provide full information about the state without previous observations. Example of that measurements could be gyroscope observation that provides an angle change, but not a result angle. This type of observations has increasing error because if there is an error on a previous step, this error affects following steps.

One more issue appears if this error is not added in considered model. In this paper model with that kind of error is presented:

$$X_{i+1} = f(X_i, U_i) * \xi_i \tag{5}$$

where X_i – a unit state, $U_i = \begin{pmatrix} \Delta x & \Delta y & \Delta\theta \end{pmatrix}^T$ – a measured shift, $\xi_i = \begin{pmatrix} \xi_x & \xi_y & \xi_\theta \end{pmatrix}^T$ – an error of measurements, $*$ – a group operation, and f – a function of two arguments $f : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$.

If it is possible to use a model presented with (5) and there exist a group $(\mathbb{R}^3, *)$, an error ξ_i could be correctly extracted. In general way it is presented with a following formula:

$$\xi_i = \left[f(X_i, U_i)\right]^{-1} * X_{i+1} \tag{6}$$

The existence of the group $(\mathbb{R}^3, *)$ provides an existence of $\left[f(X_i, U_i)\right]^{-1}$, so it is possible to calculate the error using (6).

C. Error minimization

Considered error values have a similar structure: it depends on a measurement value and a unit state. A measurement value is fixed and unit state is required to be estimated. The main requirement for a unit state estimation is to reduce error as much as possible. So now it is necessary to clarify what "reducing an error" means. Mathematically it means that it is required to find a minimum of a function that presents the error. An example of such function considered in this paper is a norm of vector:

$$\|\varepsilon(X,W)\|^2 \rightarrow \min_X \tag{7}$$

where ε – a full error, X – an unit state, W – an observation, $\|\cdot\|$ – norm of a vector.

A norm of vector could be presented in different ways. In this paper a Mahalanobis norm [10] is considered:

$$\|\varepsilon\|^2 = \varepsilon^T Q^{-1} \varepsilon \tag{8}$$

where Q – a covariation matrix of the vector ε.

III. Environment description

The focus of this paper is to fuse two types of sensors: iterative that is presented by odometry and global that is presented by scan matcher. This task appears as an issue in a SLAM problem (simultaneous localization and mapping) [11] – a task for self driving autonomous unit to orient in unknown environment and build a map of this environment using only onboard sensors. In this paper it is considered 2D laser SLAM problem [12] where laser rangefinder is used for environment observation. This sensor provides a set of points where each one of them is corresponded with a distance to an obstacle that is reached by a laser beam for every angle. Another sensor is an odometer that provides an estimation of an unit displacement and rotation.

A mobile unit has no predefined map and one hypothesis about its structure: indoor environment. So the task of sensor fusion in this case is a subtask of SLAM that provides more clear estimation of a unit pose that helps to build more correct map of an environment.

A. Laser rangefinder and map builder

A laser rangefinder provides a set of distances $\{d_i\}_{i=1}^{N}$ to obstacles for every angle generated with a step – laser scan. A geometric interpretation of these distances is presented in Fig. 1. This sensor could not track its orientation so in one time the first distance from the set could correspond to for example straight north direction and in another time orientation angle could be changed and this observation could correspond to some another direction. Some other sensors (like compass) could be used to provide orientation but they are out of the considered task.

It is required to convert observations of laser randefinder (scan) to another view that describes a unit position. One approach is to use a probabilistic grid map. A simplest example how to integrate a scan into this map – to increase in this map a probability in that cells where laser beam founds an obstacle and decrease it in free spaces.

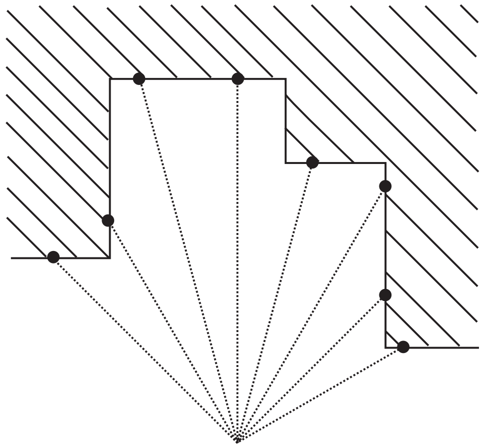


Fig. 1. Example of a laser randefinder work

Using that kind of map it is possible to estimate orientation and position of a view point of laser rangefinder and consequently a position and orientation of a mobile unit. A handler that finds the most suitable position of an input scan into the map is called scan matcher. There are many different approaches [13] how to match scans and for this research there was chosen a Monte-Carlo scan matcher [14] which randomly chooses a set of several positions of scan on a map and estimates which of them is more suitable – which doesn't break a map structure and clarify a map more correct. An algorithm of that estimation is another challenging task which requires to clarify what does it mean "more correct map that another". This algorithm is called "scan cost estimator" and an example is described in [13]. This choice of scan matcher is explained by known random parameters so it is possible to estimate a dispersion of error. That will be useful in a sensor fusion.

Using Monte-Carlo scan matcher that gets laser scan as input, it is possible to estimate a unit position. And a correspondence between a unit state and scan matcher estimation is very simple and describes with the following formula:

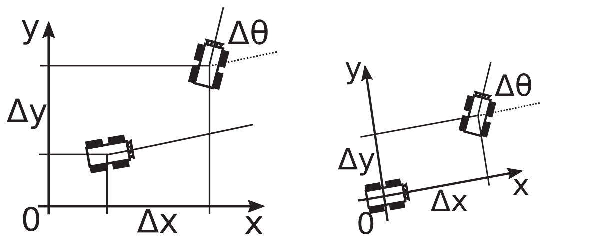
$$Z = X + \zeta \tag{9}$$

where Z – scan matcher estimation, X – real unit state, ζ – estimation error with a covariation matrix O.

B. Odometer

Odometer is a sensor that estimates a unit position using information about its movements. One of possible ways to do it – to measure amount of full circles that a wheel of a platform made. Using extra knowledge about this wheel radius it is possible to measure a velocity and a traveled distance. One of greatest disadvantage of this sensor is that it has increasing measure error – for long distance it measures with huge error, so very often odometer is used for estimate a change of a position in a short period of time. This approach doesn't reduce an error in global view but with sensor fusion it is possible to refine a state at every time moments.

Dividing a whole time period on short intervals creates multiple coordinate bases. It is possible to calculate with a little error an offset $(\Delta x \quad \Delta y \quad \Delta\theta)$ in a basis at a start position or at an end position (Fig. 2b). Or it is possible to track a position in the global coordinates if there is an observer (Fig. 2a). In this paper we consider a task when there is no external observer that could help to track a pose of a mobile unit (Fig. 2b). For example we don't use GPS because of its inaccuracy. It is said that GPS has error that is in bound of 5 meters [15]. This error is very huge when a unit in an indoor environment is tracked. Another global observer could be presented with landmarks but this approach is not considered too because the task of SLAM is to orient in unknown environment, so there is no ability to put landmarks.



(a) Global coordinates (b) Local coordinates

Fig. 2. Odometry measurements

The difference between mathematical model views presented on a Fig. 2 is described with formulas below. The first equation describes global coordinate measurements and the second – local measurements.

$$X_{i+1} = X_i + U_i \tag{10}$$

$$X_{i+1} = X_i + R_{X0Y}(\theta_i)U_i \tag{11}$$

where $X_i = \begin{pmatrix} x_i & y_i & \theta_i \end{pmatrix}^T$ – a unit state at i time moment, $U_i = \begin{pmatrix} \Delta x & \Delta y & \Delta\theta \end{pmatrix}^T$ – a measured shift, $R_{X0Y}(\theta)$ – an anti-clockwise rotation matrix of an angle θ in the $X0Y$ plate.

The equation (11) is more complex than (10) but it could be implemented in situations when there is no global observer.

There are two notices about equation (11):

1) A unit state X_i is involved as nonlinear variable because of θ_i.
2) A measurement error appears in U_i and in spite of statement 1 it could be expressed.

An existence of this expression is proved by group theory described in Section II. So the equation (11) could be updated with the $\oplus$ operator mentioned in equation (2):

$$X_{i+1} = X_i \oplus U_i \oplus \xi_i \tag{12}$$

where ξ_i – measurement error of iterative observations.

Using existence of an invert element and closure properties of a group G it is possible to extract an error value from the previous equation:

$$\xi_i = [X_i \oplus U_i]^{-1} \oplus X_{i+1}$$

where X_i – a unit state at a previous step, U_i – measurements of changes between states, X_{i+1} – a unit state at a next step.

IV. Approach definition

In previous sections we consider two types of sensors that provide global measurements (laser rangefinder and scan matcher) and iterative measurements (odometry). Scan matcher provides an estimation of a unit position and orientation as it is presented in equation (9) and odometry estimation is described like (11). These error representations correspond with common error structure mentioned in equations (4) and (6). So the task is to estimate an optimal unit pose X in a case when this estimation provides a minimum of error values in the both equations. The idea of this task was mentioned in equation (7) and the considered case could be rewritten as the following functional minimization:

$$\|\xi_i\|^2_{Q_i} + \|\zeta_i\|^2_{O_i} \to \min_{X_i} \tag{13}$$

where ξ_i and ζ_i – measurement errors from (12) and (9), Q_i and O_i – covariation matrices of these errors, $\|\cdot\|$ – Mahalanobis norm defined in equation (8).

These functional presents a quadratic form, so there exist a minimum point and this point is unique. To find this point it is possible to take a derivative for X_{i+1} from a left part from (13) and to equal it to the zero:

$$R_{X0Y}(\theta_i + \Delta\theta_i)Q_i^{-1}\xi_i - O_{i+1}^{-1}\zeta_{i+1} = 0$$

This equation is calculated using a general rule of taking a derivative of matrices:

$$((Kx+b)^T A(kx+b))'_x = K^T A(Kx+b)$$

where K, A – square matrices $K, A \in M_{n\times n}$, x, b – vectors $x, b \in \mathbb{R}^n$.

So the last point is to put theoretical values of ξ and ζ in the formula above and express X_{i+1} variable. The general formula for expression:

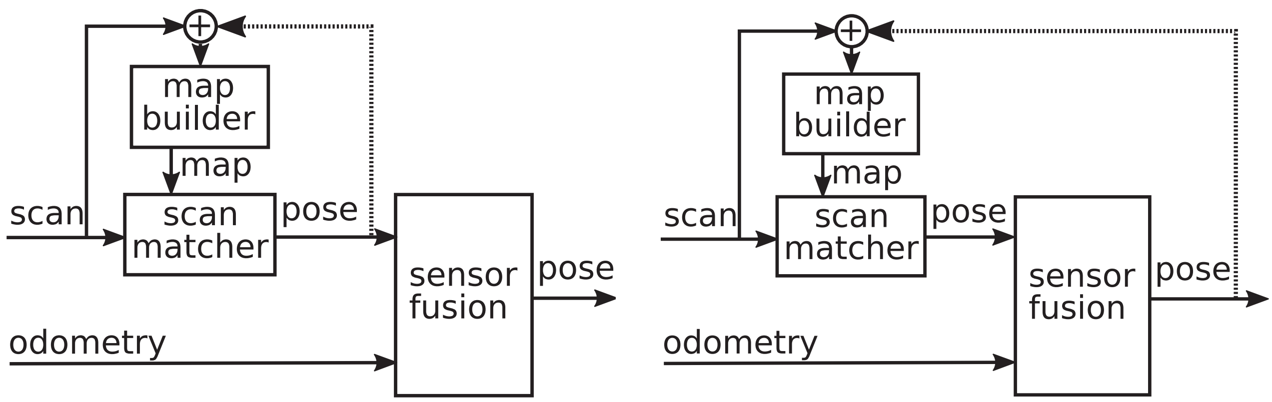
$$\begin{gathered} R_{X0Y}(\theta_i + \Delta\theta_i)Q_i^{-1}\left[X_i \oplus U_i\right]^{-1} \oplus X_{i+1} - \\ -O_{i+1}^{-1}\left(Z_{i+1} - X_{i+1}\right) = 0 \end{gathered} \tag{14}$$

where $X_i = (x_i, y_i, \theta_i)^T$ – a unit state (position and orientation on a 2D space), $U_i = (\Delta x_i, \Delta y_i, \Delta\theta_i)^T$ – an odometry offset from t_i to t_{i+1} time moments, $Z_{i+1} = \left(Z^x_{i+1}, Z^y_{i+1}, Z^\theta_{i+1}\right)^T$ – a scan macther estimation of pose, Q_i – a covariation matrix of odomery errors, O_{i+1} – a covariation matrix of scan matcher errors, $\oplus$ – operation of vector sum by direction explained in (2).

There appears the main algorithm how to fuse sensors and get a refined unit state. This algorithm is presented in alg. 1.

Algorithm 1 Algorithm of sensor fusion

1: **for** every sensor **do**
2: **if** measurements are direct **then**
3: Extract error $\xi(X)$ via equation (4);
4: **else**
5: Extract error $\varepsilon(X)$ via equation (6);
6: Build an error functional $\sum \|\xi_i(X)\|^2 + \sum \|\varepsilon_i(X)\|^2$;
7: Find a minimum by X of the functional from a step 6;



(a) Separate fusion (b) Common fusion

Fig. 3. Sensor fusion scheme

One more idea to get more clear estimation consist of a fact that now the algorithm 1 presents a case of using laser rangefinder and odometry like a scheme presented on a fig. 3a.

There two pieces of sensors data come to the sensor fusion block separately. Scan matcher gets as an input a scan and a map of an environment and returns a pose estimation, which is provided to map builder with the scan. And only after that the scan matcher pose estimation comes to sensor fusion simultaneously with an odometry pose measurement.

This current architecture could be upgraded if a map builder gets a pose refined after sensor fusion block as it is presented on a Fig. 3b. In this case map will be built more carefull, so on a next step scan matcher could provide more accurate result with less dispersion. But this improvement is specific to considered set of sensors and is not mentioned in the algorithm 1.

V. Evaluation

To test this approach MIT dataset is used [16]. The sequences presented there consist of a set of laser scans and odometry measurements, so it suits for considered approach. This dataset was used for Monte-Carlo scan matcher testing, so there are many collected results about its accuracy. Moreover this dataset has its groundtruth – a real trajectory of a mobile unit, so it is possible to estimate quantitatively the accuracy of sensor fusion. But it is important to mention that this groundtruth is built using an automatic localization method based on a known map of an environment. That means that this trajectory is not perfect and sometimes fails.

All tests are done using tinySLAM algorithm [17] which is a part of slam framework in ROS [18]. All its parameters like cell discrepancy, way of map building, estimation rule of the best position of a scan on a map are out of bounds from this topic because the main idea is to measure improves of sensor fusion application.

The results are presented in a Table I where root mean square error values are presented. These RMSE are calculated between an output unit path provided by chosen algorithm and the groundtruth trajectory.

TABLE I. RMSE values

Sequence	Length, m	Trajectory RMSE, m	
		scan matcher	sensor fusion
2011-01-19-07-49-38	68	1.280 ± 0.640	1.194 ± 0.558
2011-01-20-07-18-45	76	0.254 ± 0.045	0.237 ± 0.039
2011-01-21-09-01-36	87	0.242 ± 0.005	0.227 ± 0.004
2011-01-24-06-18-27	87	0.254 ± 0.006	0.233 ± 0.005
2011-01-25-06-29-26	109	0.260 ± 0.005	0.244 ± 0.003
2011-01-27-07-49-54	94	0.620 ± 0.030	0.586 ± 0.024
2011-01-28-06-37-23	145	2.280 ± 0.750	2.162 ± 0.693
2011-03-11-06-48-23	245	0.860 ± 0.390	0.764 ± 0.307
2011-03-18-06-22-35	80	0.103 ± 0.008	0.102 ± 0.008
2011-04-06-07-04-17	95	0.343 ± 0.025	0.307 ± 0.018
2011-10-20-11-38-39	264	5.486 ± 2.603	5.022 ± 2.247

A dispersion appears because of random component of scan matcher. Monte-Carlo scan matcher picks randomly a lot of possible hypotheses about the best position and provides that one with the greatest weight. But it could fail and do not find a good estimation, so RMSE value will be different from one launch to another.

Measurements from column 3 that is marked with "scan matcher" are collected on raw launches of tinySLAM with Monte-Carlo scan matcher. Odometry in this case is used only for prior knowledge of a neighborhood of the most suitable position that is provided to the scan matcher. So, launched algorithm has a distinction from a scheme on a Fig 3 that is in an absence of a block "sensor fusion". Measurements in column 4 are collected in a case when that block is involved in tinySLAM algorithm. So after the scan matcher provides an estimation, it is fused with odometry using a formula 14.

In the Table I there are two sequences with high values of RMSE. A result in the last line could be explained by a distance of all trajectory. It is much bigger than others, there are much more abilities for random scan matcher to fail. A result in a first line could be explained by a map of a building. There are long corridors on a way of the mobile unit, so there are a lot of places where scans looks pretty the same one to each others. In this case scan matcher could not provide valid estimation because of lack of information. Moreover in case of similar scans automatic localization with known map could fails too, so there are errors in a groundtruth too. But in case of using sensor fusion it increases accuracy and provide less error.

One more point is that sensor fusion decreases a dispersion value in all lines of Table I. It happens because of using odometry information which has not any random component from one execution to another.

An example of one test execution is presented on a fig. 4. There you can see that sensor fusion does not provide very close to groundtruth result but it refines scan matcher estimation so output trajectory becomes closer to the groundtruth.

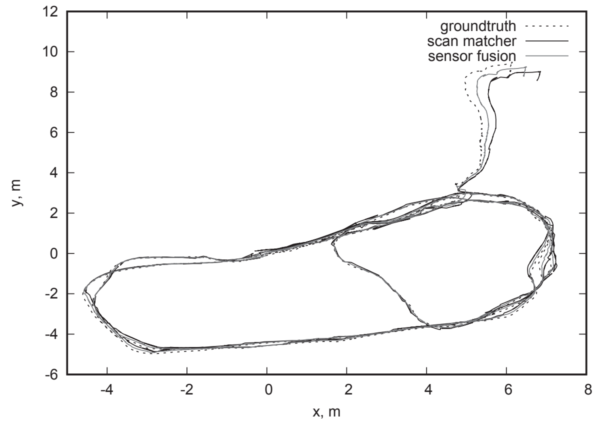


Fig. 4. Example of a test on 2011-01-27-07-49-54 sequence

This approach was tested on other sequences, for example on willow garage dataset [19]. But there is no trajectory groundtruth for that sequences and there is no ability to check an accuracy increasing. There is only one way – to estimate an output trajectory qualitatively, define where scan matcher fails and some jumps and trajectory breaks appears and define how sensor fusion handle this scan matcher behavior.

VI. CONCLUSION

In this paper the sensor fusion approach is presented. This approach could be applied for autonomous mobile units which move relying only on their onboard sensors without global observer like GPS. Two kind of sensors are considered: sensors that observe all variable directly and sensors that observes a state iteratively, based on previous observations. As an example considered in this paper there is a union of two sensors: scan matcher and odometry. So there are two types of input data: laser scan and offset of a unit in local coordinates from previous observation.

The main idea is to reduce a sum of error norms. To rich this goal group theory is used for proving an existence of estimation for a unit state. In this paper group theory helps to calculate a model dynamic where an error is not linearly added. As an example of error norm the Mahalanobis norm is used, so it calculates a weight average value of a unit state that reduces simultaneously all errors. So there is a statement that if one error has huge dispersion, an influence of the corresponded measurements on the output is very little.

The approach was quantitatively tested on MIT dataset were output trajectories were compared with a groundtruth. And there is a result that illustrates how sensor fusion refines an estimation and increases accuracy.

ACKNOWLEDGMENT

Authors would like to thank JetBrains Research for provided support and materials for working on this paper.

REFERENCES

[1] L. A. Klein, *Sensor and Data Fusion Concepts and Applications*. Bellingham, WA, USA: Society of Photo-Optical Instrumentation Engineers (SPIE), 2nd ed., 1999.

[2] W. Elmenreich, “An introduction to sensor fusion,” Research Report 47/2001, Technische Universität Wien, Institut für Technische Informatik, Treitlstr. 1-3/182-1, 1040 Vienna, Austria, 2001.

[3] R. Mur-Artal, J. M. M. Montiel, and J. D. Tards, “Orb-slam: A versatile and accurate monocular slam system,” *IEEE Transactions on Robotics*, vol. 31, pp. 1147–1163, Oct 2015.

[4] R. Mur-Artal and J. D. Tardós, “ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras,” *CoRR*, vol. abs/1610.06475, 2016.

[5] R. E. Kalman and R. S. Bucy, “New results in linear filtering and prediction theory,” *Transactions of the ASME. Ser. D, Journal of basic engineering*, pp. 95–108, 1961.

[6] A. R. Sorenson, Harold W, “Non-linear filtering by approximation of the a posteriori density,” *International Journal of Control*, vol. 8, no. 1, pp. 33–51, 1968.

[7] A. Barrau and S. Bonnabel, “Linear observation systems on groups (i).” working paper or preprint, Feb. 2018.

[8] E. Winberg, *Algebra course*. LitRes, 2017.

[9] P. Chauchat, A. Barrau, and S. Bonnabel, “Invariant smoothing on lie groups,” *CoRR*, vol. abs/1803.02076, 2018.

[10] P. C. Mahalanobis, “On the generalized distance in statistics,” *Proceedings of the National Institute of Sciences (Calcutta)*, vol. 2, pp. 49–55, 1936.

[11] H. Durrant-Whyte and T. Bailey, “Simultaneous localization and mapping: part i,” *IEEE Robotics Automation Magazine*, vol. 13, pp. 99–110, June 2006.

[12] K. Krinkin, A. Filatov, A. Filatov, A. Huletski, and D. Kartashov, “Evaluation of modern laser based indoor slam algorithms,” in *Proceedings of the 22st Conference of Open Innovations Association FRUCT*, FRUCT’22, (Helsinki, Finland, Finland), pp. 14:101–14:106, 2018.

[13] K. Krinkin, A. Filatov, A. Filatov, A. Huletski, and D. Kartashov, “The scan matchers research and comparison: Monte-carlo, olson and hough,” in *2016 19th Conference of Open Innovations Association (FRUCT)*, pp. 99–105, Nov 2016.

[14] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, “Monte carlo localization for mobile robots,” in *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No.99CH36288C)*, vol. 2, pp. 1322–1328 vol.2, May 1999.

[15] V. Naiman, A. Samoilov, and N. Iljin, *Everything about GPS-navigators*. NT Press, 2005.

[16] M. Fallon, H. Johannsson, M. Kaess, and J. J. Leonard, “The mit stata center dataset,” *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1695–1699, 2013.

[17] B. Steux and O. El Hamzaoui, “tinyslam: A slam algorithm in less than 200 lines c-language program,” in *Control Automation Robotics & Vision (ICARCV), 2010 11th International Conference on*, pp. 1975–1979, IEEE, 2010.

[18] “Slam constructor.” http://wiki.ros.org/slam_constructor. Accessed: 2019-02-11.

[19] J. Mason and B. Marthi, “An object-based semantic world model for long-term change detection and semantic querying,” in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3851–3858, Oct 2012.