Time Synchronisation and Calibration of Odometry and Range Sensors for High-Speed Mobile Robot Mapping

Fredy Tungadi and Lindsay Kleeman
Intelligent Robotics Research Centre
Department of Electrical and Computer Systems Engineering
Monash University, Victoria 3800 Australia
Email: {fredy.tungadi; Lindsay.kleeman}@eng.monash.edu.au

Abstract

Simultaneous Localisation and Mapping (SLAM) implementations are typically based upon integration of multiple sensors. One of the most common configurations for SLAM is the combination of odometry and range sensors. However, these sensors usually capture measurements at different times and thus introduce additional time synchronisation error when their data are fused together directly. As a robot moves faster, this error becomes more significant. The overall aim is to allow the robot to travel quickly and yet still be capable of obtaining accurate localisation and mapping results. This paper presents techniques for time synchronisation of multiple sensors taking into account clock drifts, inaccuracy of timestamps, and other unexpected communication and operating system delays. This paper also introduces a method of calibrating odometry and range sensors to ensure an accurate data fusion. Experimental results with and without synchronisation are shown to illustrate and validate the improvements.

1 Introduction

In the last couple of years, integration of multi-sensors has been the most popular approach for tackling the problem of Simultaneous Localisation and Mapping (SLAM) [Kleeman, 2003; Tungadi and Kleeman, 2007]. However, for any multi-sensor system, time synchronisation of sensor measurements is an important task. This is due to the fact that the sensors may sense measurements at different times. Therefore, fusing the data directly will introduce additional time synchronisation errors to the state estimation. In the robotic localisation and mapping application, this problem usually becomes more apparent when the robot moves at high speed with the SLAM performed simultaneously. As the robot moves more quickly, these errors become more significant. The aim in this study is to allow the robot to travel quickly and yet still be capable of obtaining accurate localisation and mapping results. Synchronising the times of data allows the robot to perform an accurate localisation without having to move slowly.

Most of the previous works on the time synchronisation have been in the context of networking such as Network Time Protocol [Mills, 1994]. One of the most relevant works is [Zaman and Illingworth, 2004]. This study introduces the method of time synchronisation of odometry and vision data which synchronises the times of the odometry and image data using an interval paradigm based on discrete time stamped data. However it does not take into account the clock drift and assumes that the timestamp is always accurate. In the present study, the time synchronisation method takes into account the clock drift, the inaccuracy of timestamp, and the other unexpected communication and operating system delays. The proposed method is therefore more robust and reliable.

The paper is structured as follows. The specification of sensors and robot is described in section 2, the calibration of sensors is described in section 3, the time synchronisation algorithm is described in section 4, and the minimisation of delay effect on odometry timestamp is discussed in section 5. Finally, the experimental mapping results are presented in section 6.

2 Mobile Robot Configuration and Sensor Specification

Figure 1 – Pioneer P3DX Mobile robot with front and rear Hokuyo URG laser range finders.

The robot used for experimental work is shown in figure 1

and consists of two Hokuyo URG laser range finders which are mounted on an ActivMedia Pioneer 3 DX mobile robot. The laser range finders then provide the robot with a 360 degrees field of view.

Each laser has a field of view of 240 degrees and an angular resolution of ~.36 degrees with a rotational motor speed of up to 600 RPM. The laser is provided with a 24 bit internal time source and by using the updated firmware (Ver3.1.04) and the new communication protocol (SCIP Version 2.0), a timestamp with 1 ms accuracy is attached to each laser measurement at the first sample of every scan. A more complete description of the Hokuyo URG laser range finder is presented in [Kawata, 2006] and [Kawata et al, 2005]. In order to simplify the problem of synchronisation, the two lasers' clocks are closely synchronised initially.

The robot is equipped with wheel encoders to provide odometry estimation and the measurement is sent to host computer at every ~45 ms. Unfortunately, the robot is not supplied with an internal clock, and the odometry measurements are therefore not sent to the host computer along with the timestamp. In order to allow time synchronisation, the odometry is time stamped every time the host computer receives the odometry packet from the robot. As a result, the accuracy of the timestamps is affected by transmission delay and other delays such as processing times and operating system scheduling. The calibration technique in section 5 aims to minimise the effects of these delays.

3 Calibration of Odometry and Range Sensors

In order to ensure synchronisation and data fusion, the odometry and the lasers sensors need to be calibrated. The robot employs pneumatic tyres and hence odometry calibration should be performed frequently due to tyre pressure changes. The calibration technique is an extension of the approach used by L. Kleeman [Kleeman, 2003], where both the odometry and the lasers' pose are simultaneously calibrated. This is briefly described below.

Lasers and odometry measurements are collected by observing several reliable and uncluttered features over a few minutes with the robot moving around the environment as shown in figure 2. Then, the EKF-SLAM [Davison, 1998] is then run using the collected data with relaxed data associations due to poor calibration of the odometry and the lasers. While the EKF-SLAM is run, a cost function is calculated. The cost function is the sum of square differences between expected measurements and actual measurements based on average feature position and odometry. The cost function is then minimised with respect to each parameter serially using Brent's optimisation method [Press et al, 1992]. As in [Kleeman, 2003], the parameters and the order of optimisation are chosen to decouple their effects on the cost function minimisation. The parameters in order of optimisation are as follows:

$$R = \frac{R_r + R_l}{2}$$

$$RonB = \frac{R}{B}$$

$$delta = \frac{R_r - R_l}{R_r + R_l}$$

$$\begin{aligned} \alpha_{sum} &= \alpha_{front} + \alpha_{back} \\ \alpha_{diff} &= \alpha_{front} - \alpha_{back} \\ x_{lsum} &= x_{lfront} + x_{lback} \\ y_{lsum} &= y_{lfront} + y_{lback} \\ x_{ldiff} &= x_{lfront} - x_{lback} \\ y_{ldiff} &= y_{lfront} + y_{lback} \end{aligned} \quad (1)$$

where R_r is the radius of right wheel, R_l is the radius of left wheel, B is the wheel separation, while x_{lfront}, y_{lfront} and α_{front} are the pose of the front laser with respect to the robot coordinate frame, and x_{lback}, y_{lback} and α_{back} are the pose of the back laser with respect to the robot coordinate frame.

As an example of decoupling, optimising $RonB$ will only affect the odometry turning angles and optimising $delta$ will straighten curved odometry paths when the robot travels in a straight line. A similar concept applies to the remaining parameters.

Figure 3 shows the evolution of the cost function using the Brent's optimisation method. As depicted in figure 3, after several iterations, the cost is reduced to a minimum value leaving the odometry to just involve non-systematic error in its estimation of the robot's pose. The spikes in figure 3 show the result of bad guesses of the parameters taken by the Brent's optimisation method.

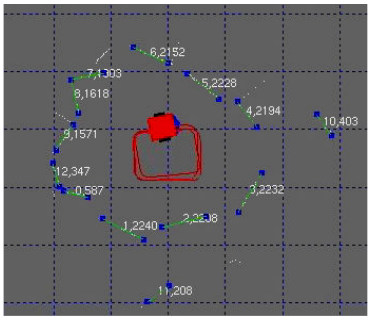


Figure 2 - The example environment for calibration of odometry and lasers. The grid size is 1 m.

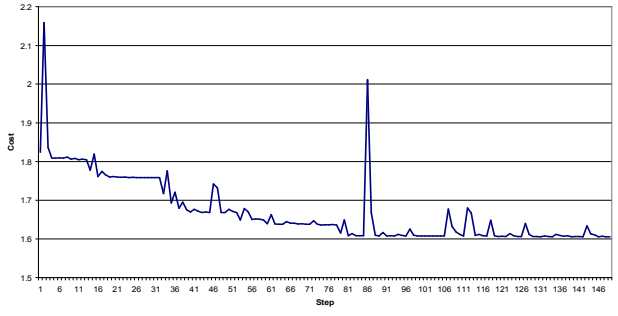


Figure 3 - Calibration cost evolution using Brent's optimisation approach.

4 Time Synchronisation Algorithm

4.1 Clock Synchronisation Problem

Computer clocks like all clocks are never in perfect agreement. Let C_l and C_{pc} be clock times on the laser and host computer respectively. The assumed properties of clocks related to time synchronisation are as follows:

1. Clock offset (θ): the difference between the times on two clocks,
$$\theta = C_{pc} - C_l \quad (2)$$
2. Clock drift (ρ): the counters of the two clocks operate at slightly different frequencies. The drifting of the clocks depends on their quality, operating temperature, and other factors such as the battery voltage.

Assuming the clocks (host computer clock and laser clock) are running at the same frequency, then the clock offset, θ should always remain constant at any time. However, as mentioned above, in reality, they are running at different frequencies. The plot in figure 4 is an example where a constantly decreasing in offset behaviour is observed. A Similar behaviour is also observed in [Carballo et al, 2007].

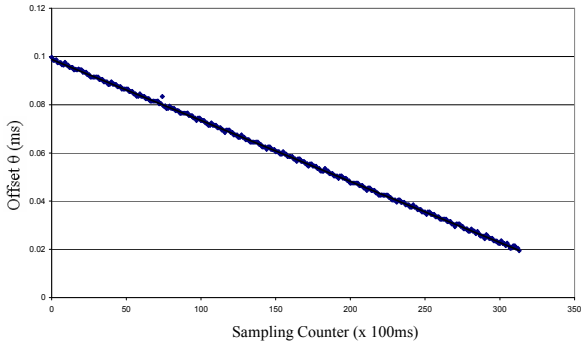


Figure 4 – Offset in milliseconds versus sensor time (1 sampling count is equivalent to 100 ms).

4.2 Time Synchronisation between Lasers and Host Computer

The method described in this section is adapted from [Carballo et al, 2007]. In order to describe the laser local clock with respect to the host computer clock, the following equation is used:

$$C_{hl}(t) = (1+\rho)C_l(t) + \theta \quad (3)$$

where C_{hl} is the clock value of laser in the host computer time reference frame and C_l is the clock value of laser in its local reference frame. Therefore, the time synchronisation is simply a measurement of both the time offset, θ and the relative drift rate, ρ.

The measurement of offset, θ is based on Cristian's algorithm [Cristian, 1989] as in [Carballo et al, 2007]. As depicted in figure 5, there are four clock values used to calculate the offset, where $C_{pc}(t_1)$ and $C_{pc}(t_4)$ are the time of sending and receiving of time request respectively. Because the laser only sends back one value of timestamp, it is assumed that $C_l(t_2) = C_l(t_3) =$ timestamp and the offset, θ is then calculated as follows:

$$\theta = -C_l(t_2) + \left(\frac{C_{pc}(t_4) - C_{pc}(t_1)}{2} \right) \quad (4)$$

The effect of the clock drift between laser and host computer clock can be reduced by fitting a straight line equation to the plot in figure 4 in order to estimate future drift values where the gradient of the line equation is used to adjust the drift between the two clocks. As reported in [Carballo et al, 2007], the effects of clock drifts, although reduced, are rather difficult to estimate and to cancel by just one attempt due to the changes in clock drift as described in section 4.1. In order to maintain proper synchronisation, periodic adjustments are necessary.

Through a series of experiments we found that in order to maintain the accuracy of the time synchronisation in the order of 1 ms, the adjustment has to be done periodically at every ~3 minutes and the results are shown in figure 6.

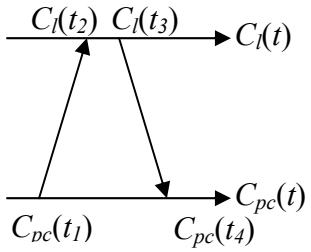


Figure 5 - Cristian's algorithm

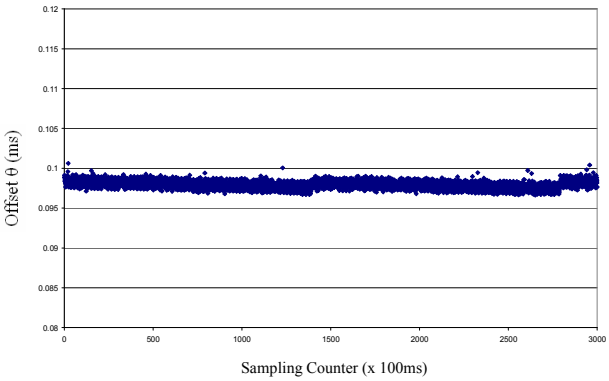


Figure 6 - sensor time vs. offset with periodic drift adjustment.

4.3 Synchronisation between Lasers and Odometry

Laser data and odometry data are sampled at different times as shown in figure 7. The reason is because they are very likely to start at different times and sampled at

different rates. Given the offset of the timestamps at any time, the method of odometry-laser synchronisation here aims to minimise the effect of interval offset when fusing data. One way to achieve this is by using linear interpolation of odometry to estimate the value of the robot's pose at the time when the laser scans are captured. The odometry has been chosen to be interpolated instead of the lasers since it has fewer variables compared to the laser where the interpolation will involve interpolating the whole scans. The interpolation of odometry is calculated as follows:

$$X(t_n) = X(t_{o1}) + \left(\frac{X(t_{o2}) - X(t_{o1})}{t_{o2} - t_{o1}} \right) (t_n - t_{o1}) \quad (5)$$

where t_n is the most recent sampling time of laser, t_{o2} is the most recent sampling time of odometry, t_{o1} is the second most recent sampling time of odometry, and X is the robot's pose which is expressed with respect to time.

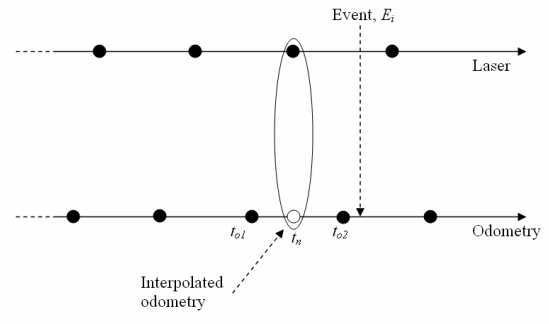


Figure 7 - Timelines showing discrete periodic timestamps.

4.4 Compensation of the Effects of Delays across the Laser Scans.

Section 2 describes that there is only one value of laser timestamp per scan contain multiple laser measurements. Thus, the process of interpolation of odometry to synchronise with laser measurement using this timestamp will only synchronise the odometry with the first sample of the laser scans. With 240 degrees field of view and 600 RPM rotational motor speed, the maximum measurement lag from the beginning to the end of scan is ~66.7 ms in the direction of rotation meaning that some of the scan points especially near the end of scans are not accurately synchronised with the odometry.

The first step in tackling this problem is by attaching each step in the scans a timestamp. Knowing the value of the timestamp at the beginning of the scans and the motor speed, the time at each step of the scans can be estimated using the following equation:

$$t_s(s) = t_{s0} + s * \Delta t_s \quad (6)$$

where t_s is the timestamp of each individual scan point, s is the step number of the scans, t_{s0} is the timestamp of step 0, and Δt_s is the timestamp difference between consecutive scan points which is worked out to ~0.1ms with rotational motor speed of 600 RPM.

The next step can be explained as follows. Each of individual scan point is captured with respect to the pose of the laser and since they have different relative delay, when the robot moves, these scan points are captured from different laser's pose as illustrated in figure 8. Thus, they should not be treated as if they are captured from exactly the same position. Given that the odometry is already time-synchronised with the first sample of the scan and the current robot's rotational and translational velocity are known from odometry, the laser's pose at each individual timestamp can be estimated. The scans are then transformed back to a reference pose (in this case, the reference pose is the pose at t_{s0} since it is the time where odometry and the laser is synchronised) by using equation 7. The scan points can be transformed back to its original polar format using scan projection and linear interpolation method if needed (see [Diosi and Kleeman, 2007] for more detail).

$$T_{LD}^{s} = \begin{bmatrix} \cos(-\omega s \Delta t) & -\sin(-\omega s \Delta t) & -V_x s \Delta t \\ \sin(-\omega s \Delta t) & \cos(-\omega s \Delta t) & -V_y s \Delta t \\ 0 & 0 & 1 \end{bmatrix}$$

$$T_L = \begin{bmatrix} \cos(\theta_L) & -\sin(\theta_L) & x_L \\ \sin(\theta_L) & \cos(\theta_L) & y_L \\ 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$\hat{p}_s = T_L^{-1} T_{LD}^{s} T_L p_s$$

where T_L is the homogenous transformation matrix that transforms the scan points from laser coordinate frame to robot coordinate frame, T_{LD}^{s} is the homogenous transformation matrix that transforms the robot's pose with respect to the reference robot's pose frame, s is the step number of the scan point, p_s is the scan point at s^{th} step, ω is the rotational velocity of the robot, V_x is the robot's velocity in the x direction and V_y is the robot's velocity in the y direction.

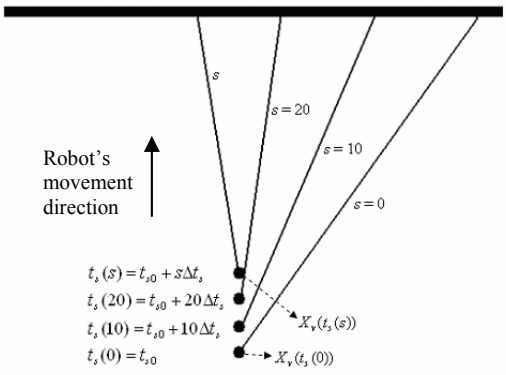


Figure 8 – Illustration of the effects of motor delay effects on the laser scans, where each scan point is measured with respect to the laser's pose.

5 Minimising Delay Effect on Odometry Timestamp by Calibration

This section presents a simple calibration procedure adapted from a method by G. Taylor and L. Kleeman [Taylor and Kleeman, 2006] that can compensate for the latency between the lasers and the odometry caused by transmission delay and other delays such as processing delays.

The calibration technique is based on minimising the hysteresis induced by the delay for some cyclic motion. Here, the robot is rotating left and right while the encoder and the laser readings are recorded along with a timestamp. It is also worth mentioning that in order to reduce the effect of slippage on odometry, this calibration procedure was conducted on a hard-floor with a reasonably slow rotational speed of 20 degrees per second. Both of the readings are then used to calculate the orientation of the robot. Encoder orientation estimation is simply calculated using the odometry equation as in [Kleeman, 2003] while one of the extracted laser lines is tracked and the orientation changes provide estimation of the robot's orientation. Let $x(t)$ be the estimated robot's orientation provided by the laser and $e(t)$ be the estimated robot's orientation provided by the wheel encoder.

Figure 9 shows the plot of estimated orientation using both the odometry and the extracted laser line versus time resulting from the cyclic motion. Figure 10 shows the orientation value of the extracted line x(t) plotted against odometry orientation value e(t), with a visible hysteresis loop resulting from the lag between these two variables before applying the time synchronisation algorithm described in previous sections. Figure 11 shows an improvement of hysteresis after applying the time synchronisation algorithm. The loop can be closed by phase shifting the encoder values by the acquisition delay, δt. Therefore, the latency δt can be calculated as the phase shift that minimises the residual error between extracted laser line orientation value x(t) and encoder orientation value e(t + δt). The phase shift is applied using linear interpolation to calculate the encoder orientation e(t + δt). Figure 12 shows the residual error versus latency, δt for phase shifts in 5ms increments between $\pm$ 100 ms.

The plot in figure 12 exhibits minima at about 12 ms, indicating that the encoder is sampled about 12 ms before the laser line is extracted. After compensating this delay, the hysteresis between the two variables has disappeared as shown in figure 13.

The accuracy of the method described here depends very much on the accuracy of the timestamps. Since the odometry is time stamped by software at the time where the host computer receive the packet, it is necessary to make sure this timestamp is not affected by operating system's context switching delays. For demonstration of this problem, interrupts of the packet receiver process are intentionally introduced and the effect of the delays is shown in figure 14. In order to tackle this problem, a line is fitted to the plot in figure 14 excluding the outliers. The arrival time which is the lowest from the fitted line, exhibits minimum transmission delay. The fitted line is then shifted downwards such that it is equal or below all the arrival times. Finally, this line is used to estimate the timestamp whenever the timestamps arrive longer than expected.

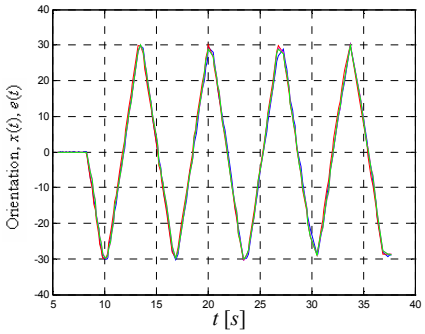


Figure 9 – extracted laser line orientation value $x(t)$ and odometry orientation value $e(t)$ versus time t.

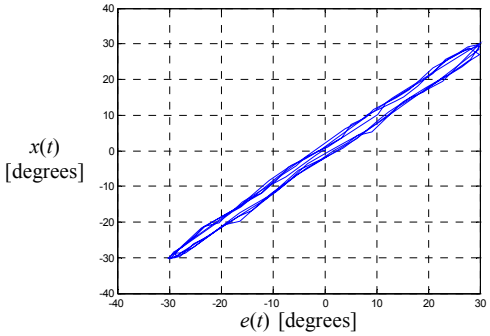


Figure 10 - Hysteresis due to acquisition delay between corresponding extracted laser line and encoder orientation measurements.

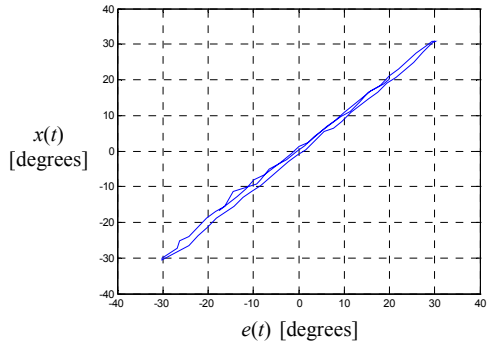


Figure 11 - Relationship between extracted laser line and encoder measurements after synchronisation.

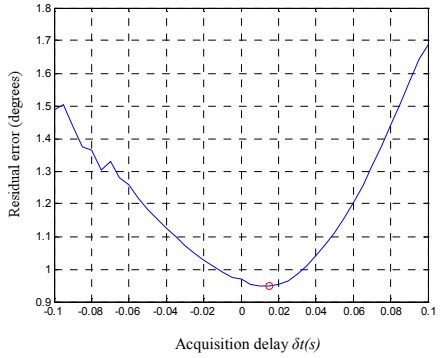


Figure 12 - Residual error for estimation of time delay from linear regression.

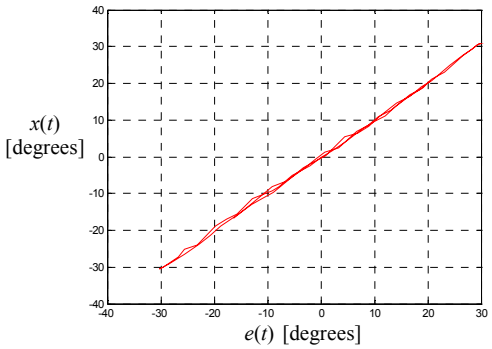


Figure 13 - Relationship between extracted laser line and encoder measurements after compensating for latency.

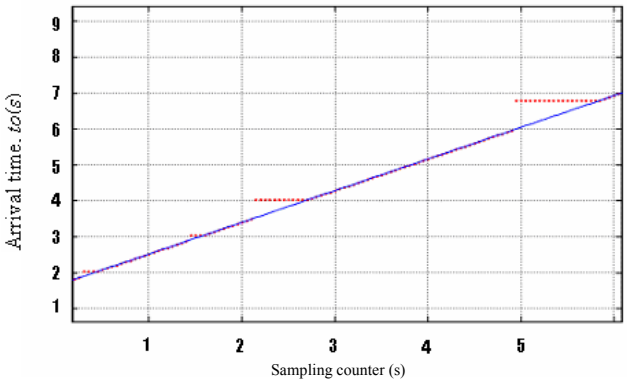


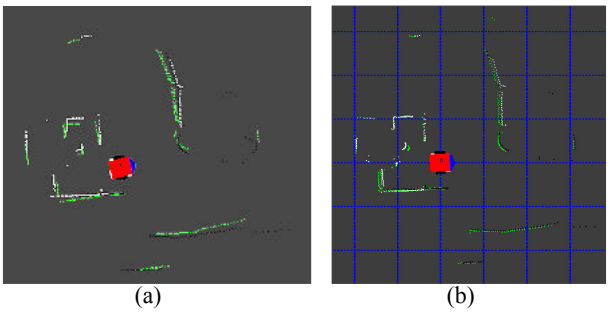
Figure 14 – Plot of expected arrival times versus actual arrival times. The red dot in the plot shows the effect of context switching delay.

6 Experimental Mapping Results

Experiments of SLAM were carried out to verify the feasibility of the time synchronisation and calibration algorithm. The implementation of SLAM is based on previous work [Tungadi and Kleeman, 2007] where the SLAM is implemented using Extended Kalman Filter. The prediction model is based on odometry model in [Kleeman, 2003] and the measurement process is performed using Polar Scan Matching [Diosi and Kleeman, 2007].

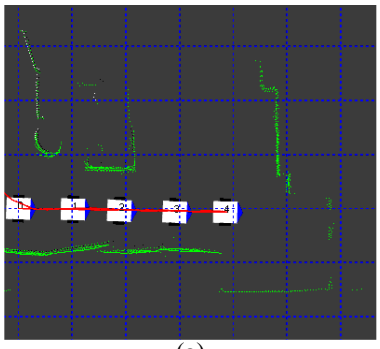
Two experiments were carried out to measure the accuracy of integrating time synchronisation in the SLAM implementation. In the first experiment, the robot rotates left and right on the spot at ~30 degrees per second and then moves in a straight line at 1000 mm/s. When the robot rotates left and right, without time synchronisation the alignment error between current and reference scans is clearly larger than the synchronised experiment as shown in Figure 15. By integrating the time synchronisation, the average error was reduced from 5 degrees to 2 degrees. However when the robot was moving in a straight line, the difference in error is not as obvious (shown in figure 16). The average error is reduced from 10 cm to 6 cm.

In the second experiment, the robot travels continuously at an average speed of 500 mm/s (except when turning) in a small loop-like environment where the scan-matching and the SLAM update are updated on-the-fly. With sensors calibration and time synchronisation, the measurement innovations produced by Kalman filter reduced from an average of 0.05m to 0.03m in translational innovation and the average of 4 degrees to 2 degrees in rotational innovation. It is also worth mentioning that without sensors calibration and time synchronisation, when the robot is prior to closing the loop, the measurement innovation is as big as ~0.25 m in translational innovation and ~5 degrees in rotational innovation. However, when the time synchronisation is integrated into the SLAM, the measurement innovation prior to closing the loop is reduced to ~0.1 m in translational innovation and ~3 degrees in rotational innovation.

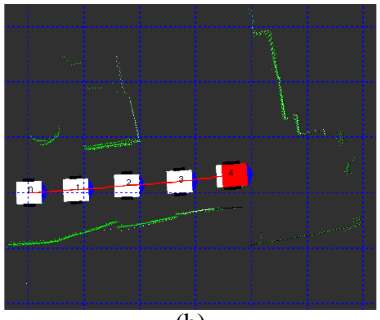


(a) (b)

Figure 15 – Rotation on the spot experiment. The left picture shows the experimental result of mapping without integration of time synchronisation and the current scans are not accurately aligned with reference scans. The right picture shows the experimental result of mapping with integration of time synchronisation and the alignment between the current and reference scans is more accurate.

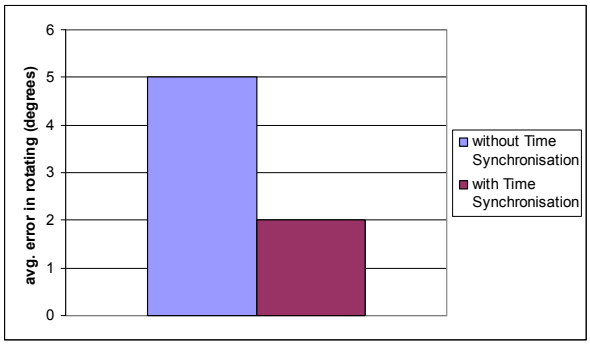


(a)

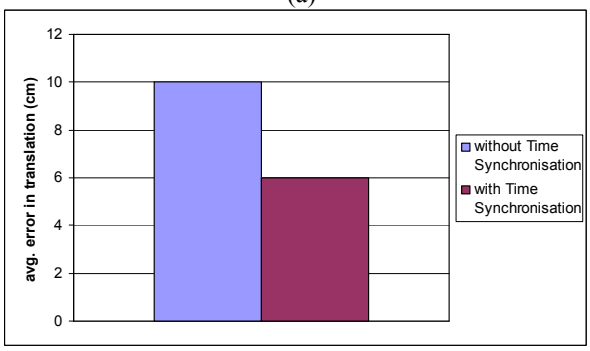


(b)

Figure 16 - Straight line motion experiment. The top picture shows the experimental result of mapping without integration of time synchronisation whereas the bottom picture shows the experimental result of mapping with integration of time synchronisation. The results are both visually accurate, although when compared with the ground truth (tape measurement) the result of mapping with integration of time synchronisation is slightly better.



(a)



(b)

Figure 17 - The top bar diagram shows the comparison of the average error in rotation and the bottom bar diagram shows the comparison of the average error in translation.

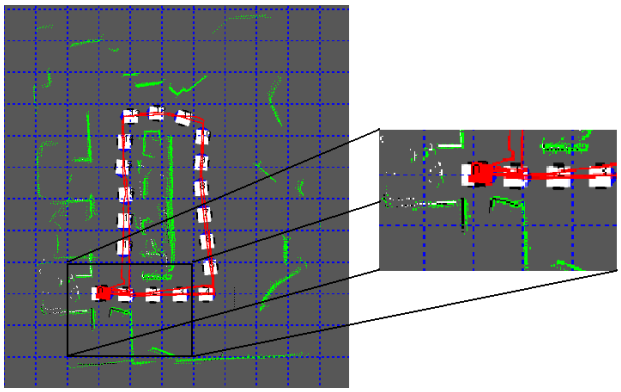


Figure 18 - the map result of a small loop environment (with time synchronisation). The grid size is 1 m.

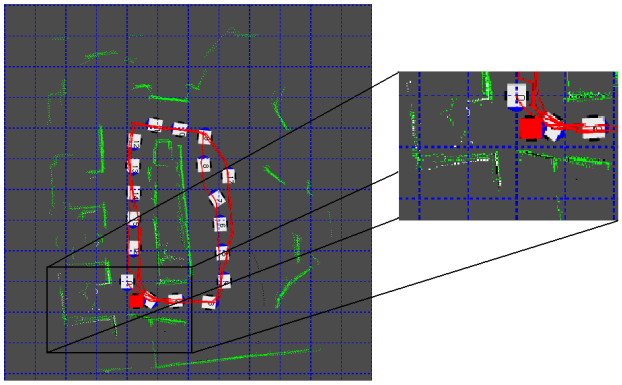


Figure 19 - the map result of a small loop environment (without time synchronisation). The grid size is 1 m.

7 Conclusions

A method of calibration and time synchronisation between odometry and range sensors as well as its importance has been described in this paper. The method consists of:

- Calibration of odometry and range sensors poses with respect to the robot.
- Synchronisation of laser clock and host computer clock.
- Synchronisation of odometry and laser data based on their timestamp.
- Compensation of the delay effects across the laser scans.
- Elimination of various types of delays such as processing delay, transmission delay and context switching delay.

This method has been shown to successfully synchronise the data from the odometry and the lasers based on their timestamps. Nevertheless, the methods described in this paper only synchronise two sensors. The future work will be generalised to synchronising any number of sensors.

Future work will also involve the application of this method to other robotics tasks such as high-speed obstacle avoidance and high-speed robotic exploration.

8 Acknowledgment

The financial support of the ARC Centre for Perceptive and Intelligent Machines in Complex Environments is gratefully acknowledged.

References

[Carballo et al, 2007] A. Carballo, Y. Hara, H. Kawata, T. Yoshida, A. Ohya, S. Yuta, “Time synchronisation between SOKUIKI sensor and host computer using timestamps”, Proceedings of the JSME Conference on Robotics and Mechatronics ROBOMEC 2007, Akita, Japan, 2007, Paper 1P1-K05.

[Kleeman, 2003] L. Kleeman, “Advanced sonar and odometry error modeling for simultaneous localisation and map building”, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Las Vegas 2003, pp 699-704.

[Tungadi and Kleeman, 2007] F. Tungadi and L. Kleeman, “Multiple laser polar scan matching with application to SLAM”, Proceedings of the Australasian Conference on Robotics and Automation, Brisbane, Australia, 2007.

[Kawata, 2006] H. Kawata, “URG series communication protocol specification SCIP-version 2.0”, Drawing No. C-42-03320B, Oct 2006. Available: http://www.hokuyo-aut.jp

[Kawata et al, 2005] H. Kawata, A. Ohya, S. Yuta, W. santosh, T. Mori, “Development of ultra-small lightweight optical range sensor system”, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Alberta, Canada 2005, pp 1078-1083.

[Press et al, 1992] W. H. Press, S.A. Teukolskly, W.T. Vetterling, B.P. Flannery, “Numerical recipe in C: the art of scientific computing”, 2nd edition, Cambridge

University Press, 1888-1992, chapter 10.

[Cristian, 1989] F. Cristian, "Probabilistic clock synchronisation", Distributed Computing, Vol. 3, pp 146-150, Springer-Verlag 1989.

[Taylor and Kleeman, 2006] G. Taylor and L. Kleeman, "Visual perception and robotic manipulation", STAR, Vol. 26, pp 195-197, Springer-Verlag Berlin Heidelberg 2006.

[Diosi and Kleeman, 2007] A. Diosi and L. Kleeman, "Fast laser scan matching using polar coordinates", International Journal of Robotics research, Vol. 26, pp 1125-1153, October 2007.

[Zaman and Illingworth, 2004] M. Zaman and J. Illingworth, "Interval-based time synchronisation of sensor data in a mobile robot", Proceedings of the Sensor Networks and Information Processing Conference, 2004, pp463-468, 14-17 Dec. 2004.

[Mills, 1994] D. L. Mills, "Internet time synchronisation: the network time protocol", Global States and Time in Distributed Systems, IEEE Computer Society Press, 1994.

[Davison, 1998] A. Davison, "mobile Robot Navigation Using Active Vision". PhD thesis, University of Oxford, 1998.