UWB based Real-Time Cooperative Localization System

ST Engineering-NTU Corporate Lab
Nanyang Technological University, Singapore

Abstract—Ultra-wideband (UWB) based positioning is preferred in multipath and non-line-of-sight (NLoS) conditions, especially under indoor environments. In this paper, we present the work-in-progress to establish a real-time localization system based on UWB ranging. A hardware board based on UWB-Impulse Radio (UWB-IR) transceiver and a ranging scheme is developed to actively carry out range measurements between the peer nodes. Once a range is obtained, it goes through a NLoS detection scheme before being fed to a positioning algorithm. The positioning algorithm is initialized by achieving a relative coordinate system and followed by refinement of the coarse location estimates using non-linear least-square minimization. Data from inertial navigation sensors is to be fused with the UWB positioning to improve accuracy and update rate. Tests in different environments have been conducted to validate the performance of UWB ranging and positioning algorithm.

I. INTRODUCTION

Due to the precise localization ability, indoor positioning has gained attention for many beneficial applications such as defense, rescue, logistics, medical, gaming and entertainment. However, conventional positioning systems such as GPS are unreliable in indoor environments. UWB technology comes in as an ideal alternative solution to provide localization in indoor environments. UWB technology performs well under multipath interference because of the sub-nanosecond pulse width. Multipath components can be clearly distinguished from the first path; hence accurate estimation of the Time of Arrival (ToA) of the incoming signal is possible.

A hardware board with a half-duplex UWB-IR transceiver is presented by POZYX (https://www.pozyx.io/). This board is intended to be controlled by a set of commands exerted from an external platform such as Arduino. This design has segregated the positioning algorithm (from the ranging scheme running on the on-board STM32 microcontroller) to run on the external platform allowing the required speed and computational power to be maintained properly. An Asymmetric double sided two-way-ranging (TWR) scheme is presented in Decawaves implementation [1]. Considering an anchor based system, this scheme has been extended to a more efficient scheme by cutting down the number of messages required for a peer-to-peer ranging. The usage of a full-duplex UWB-IR transceiver opens up the opportunity to explore more efficient and fast ranging schemes. Ranging schemes based on minimal messaging and fast data rates are another way of improving the ranging time.

II. SYSTEM OVERVIEW

The application scenario involves a number of nodes which are mobile. The nodes will move over a larger area and there may be blockages between some of the nodes. The cooperative localization is to provide local position of all the nodes in 3D.

The platform shown in Figure 1 is based on Decawave UWB-IR transceiver IC (DW1000) which has a coherent receiver with sub-nanosecond time resolution ToA circuit and 1ns channel impulse response sampling. Both simultaneous ranging and data communication are through UWB-IR and has a power Consumption of around 870mW. The control of the Decawave UWB-IR transceiver is from the STM32 microcontroller through the SPI port. Peer-to-peer ranging algorithm and data transfer with the UWB-IR transceiver IC is implemented in firmware. Cooperative position is done on BeagleBone Black (BBB) which has a power consumption of around 1W to 2.3W. The peer-to-peer range and data are exchanged with the BBB through UART port by USB interface. Linux operating system is used in the BBB with co-operative positioning and control implemented in C and Python. To keep the cost low, all nodes use independent low cost clock sources of 10ppm.

Fig. 1: Localization system

III. HARDWARE

Initial prototyping of the system was carried out with the Decawave TREK1000 Evaluation kit which is again based on DW1000. The DW1000 is a half-duplex transceiver. The use of multiple transceiver ICs controlled by a single STM32 MCU is one way of achieving full-duplex transmission. A new hardware board with 3 transceivers is under development. This board is developed in two versions, one consisting of the
DW1000 IC and the other consisting of DWM1000 module. Figure 2 shows the TREK1000 and the developed board with multiple transceivers.

Figure 2 shows the TREK1000 and the developed board with

Figure 2: Hardware circuits

Having a number of UWB-IR transceivers, enables the possibility of utilizing wireless time synchronization over the UWB channel using one of the transceivers while doing peer-to-peer ranging using other transceivers. Furthermore, various TDoA schemes can also be explored and implemented to achieve reduced timing and better accuracy for peer-to-peer ranging. Inertial Measurement Unit (IMU) is integrated into the system to track the position and orientation of the agents based on the 9-axis motion tracking device MPU9250 that combines a 3-axis gyroscope, 3-axis accelerometer and a 3-axis magnetometer. Additionally a pressure sensor MPL3115A2 is also added to the hardware to obtain altitude information. The information from these two sensors can be fused with the UWB based positioning to increase accuracy and update rate.

IV. RANGING SCHEME

Three messages exchange is used to obtain the peer-to-peer range between two nodes\(^1\) followed by a broadcast message to inform all nodes of the calculated range. Node 1 first sends a poll message to pair with node 2 which replies with a response message after a processing delay of \(T_{\text{reply1}}\). Node 1 uses a timer to measure the ToA of the response message from node 2 and compute the time difference between the time it sent the poll message to the time it receives the response message, known as \(T_{\text{round1}}\). All nodes in the system uses independent clock with 10ppm accuracy. Due to this clock drift and offset with respect to each node, the value of \(T_{\text{reply1}}\) has an error if used directly in node 1 to compute the Time of Flight (ToF), \(T_{\text{prop}}\). This issue is resolved by node 1 sending a final message to node 2 after a processing delay \(T_{\text{reply2}}\). Node 1 transmits its measured \(T_{\text{round1}}\) and \(T_{\text{reply2}}\) in the final message to node 2. Node 2 uses a timer to measure the ToA of the final message from node 1 and compute the time difference between the time it sent the response message to the time it receives the final message, known as \(T_{\text{round2}}\). The peer-to-peer range is computed based on the ToF, \(T_{\text{prop}}\) between nodes 1 and 2, at node 2 using the mean of the measurement rounds given by [1]:

\[
\hat{T}_{\text{prop}} = \frac{T_{\text{round1}} \times T_{\text{round2}} - T_{\text{reply1}} \times T_{\text{reply2}}}{T_{\text{round1}} + T_{\text{round2}} + T_{\text{reply1}} + T_{\text{reply2}}}
\]

After node 2 computed the peer-to-peer range using (1), it transmits the computed range to all nodes using a broadcast message. Every message starts off with a preamble, Start of Frame Delimiter (SFD) and followed by data which vary in length according to number of data bytes to be transmitted. The duration can be computed as given in the equation for \(x\) in the Figure 3. If the data rate is chosen to be 110kbps, a typical total time for the 4 messages is around 10.6 ms. For \(m\) nodes in the network, \(4[m(m - 1)/2]\) messages are required to get the peer-to-peer ranges between all nodes. The measured time for 4 nodes in the network is around 80mS including the time required to transfer the peer-to-peer ranges to the BBB Linux user space through the UART connection over USB from the STM32. Timers are used to abort the task if the peer-to-peer range among nodes are not returned at each message.

The data rate could be increased such that the size of the ranging frames are reduced to several hundred micro seconds. Hence the peer-to-peer ranging time can be brought down to 2.5 ms/pair. Based on the above timing, ranging update rate is just above 50Hz for a 4 nodes cluster. The timing of the different messages for the 850kbps data rate is as shown in Figure 3.

\[
x (\mu s) = \left[ 20 + 8 (n + 12) \right] / 0.05
\]

\(n\) : number of data bytes

Fig. 3: Timing diagram for 850kbps data rate

In order to minimize the delay caused by the back and forth communication between STM32 and BBB, A coordination scheme is implemented such that once commanded with the group of node addresses, Decawave can automatically run the peer-to-peer ranging between all the pairs of nodes.

A. Firmware implementation

The firmware implementation is based on the STM32F105RC, ARM Cortex M3 microprocessor. The firmware has a list of functions that can be used to command the STM32. The commands are passed via USB either from

\(^1\)As we are developing an anchor-free positioning system, the terms "Node" and "Agent" are interchangeable.
the Laptop (using Matlab or Python) or can be controlled by the BBB. Once a command is executed, STM32 will return COMMAND_SUCCESS or COMMAND_FAIL, back to BBB after the execution of the command. The main commands used by the BBB are DECA_DO_RANGING, DECA_GETRANGEINFO and DECA_GETPOSINFO. DECA_DO_RANGING function initiates the ranging operation and depending on the arguments passed along with the command, the coordination may or may not be controlled by Decawave. If the coordination is selected, Decawave will perform ranging between all the possible ranging combinations and store the range information in the range table. DECA_GETRANGEINFO function returns the range table info stored and once this command is issued, the ranging table will be flushed from the STM32 memory. This function does not initiate any ranging operation. Range information will be available as a result of calling DECA_DO_RANGING directly. The firmware also offers an additional DECA_GETPOSINFO command that provides the Least Square Estimate (LSE) of the nodes position. Using this position information from the firmware will allow the BBB to avoid the initialization step and carry out only the iterative positioning.

An additional sequence message is added to the scheme in order to share the time slot information to all the nodes in the cluster. The node 1 (assuming the initiator) will send the sequence message to all the other nodes. For this scheme, a peer-to-peer ranging takes about 2.5 ms for 850kbps data rate. Based on this timing, the TIMER interrupts have been employed. So, for example, if there are 5 nodes in a cluster, after 2.5ms×5 (i.e., after node 1 ranges with nodes 2, 3, 4 and 5), node 2 will be interrupted to initiate the ranging with nodes 3, 4 and 5 and so on. The TIMER-based interrupts have been employed, so as to avoid any collapsing of the scheme due to unavailability or failure of any of the nodes in the cluster. Currently, the firmware allows up to 9 nodes to be in a cluster. If necessary, this can further be increased by making slight modifications.

Apart from the range information, some quality parameters such as first path power (FP power), total received power (RX power) and standard deviation of noise are provided for the validation of the calculated ranges. To cut down the number of messages used, a novel form of multiway ranging scheme is under development. The new scheme relies on broadcast messaging and expected to reduce the ranging time by around 50% of the conventional scheme discussed earlier. The messaging sequence and proper time stamping is found to be crucial for the implementation of the new scheme.

V. LOS, NLOS AND NLO S DETECTION

Due to possible blockage between nodes, it is important to identify which measured peer-to-peer ranges are in line-of-sight (LoS) and which ranges are in NLoS. We break down NLoS into two categories: 1) the obstacle can block the radio wave; 2) the obstacle block the visual path between two nodes but not the radio wave. We term the second NLoS as optical non-line-of-sight (ONLoS). In case of true NLoS where the radio wave is blocked, the ranging measurements should be discarded. In case of ONLoS, the ranging measurements will be positively biased because velocity of propagation through the obstacle is slower than that in free space. It’s not a easy task to distinguish between ONLoS and LoS because their channel impulse responses are very similar. As direct path power is absorbed and reflected by the ONLoS/NLoS obstacle, the first path power will be much weaker than the power in the free space i.e. LoS. Motivated by this observation, we propose to identify ONLoS/NLoS based on the difference between the total received power and the first path power, both of which are accessibile from DW1000 as a quality parameter for reception.

Figure 4 shows the measured RX power and FP power for the different propagation environments. It can be observed that the ONLoS and NLoS RX power exhibits large deviation from the FP power as well as the free space power at the measured distance. The power difference in dB between the RX power and FP power is computed and compared with a threshold of 6dB. If the difference is below 6dB, we consider it as LoS; otherwise ONLoS/NLoS. The complete detection scheme is detailed in [2].

![Fig. 4: RX power and FP power in different environments.](image)

VI. POSITIONING MODULE

We are investigating and implementing infrastructure-less i.e. anchor-free cooperative localization. Cooperative localization enables inter-agent communication, which is not activated in the non-cooperative localization, thus requiring less anchor density.

A. Initialization

Consider a case of four agents and their radio connections are denoted by the dashed lines in Figure 5. At the initial phase, we establish a relative coordinate system (we need it because no anchors are available) by

1) assuming node 1 is located at origin i.e. \( p_1 = [0, 0]^T \);
2) placing node 2 on positive \( x \)-axis and its coordinates is \( p_2 = [d_{1,2}, 0]^T \) where \( d_{1,2} \) is the range measured between node 1 and node 2;
3) defining positive $y$-axis by node 3 whose location $p_3 = [x_3, y_3]^{T}$ can be determined as

$$\hat{x}_3 = d_{1,3} \cos \theta_{213},$$

$$\hat{y}_3 = d_{1,3} \sin \theta_{213},$$

where $\theta_{213} = \arccos \left( \frac{d_{1,2}^2 + d_{1,3}^2 - d_{2,3}^2}{2d_{1,2}d_{1,3}} \right)$ denotes the angle between $d_{1,2}$ and $d_{1,3}$. Assuming that the agents are fully connected and defining $\ell \in N_k$ as the neighbors of node $k$, available now are the estimates of nodes 1,2 and 3, the remaining nodes’ locations may be estimated as

$$(x_k - \hat{x}_\ell)^2 + (y_k - \hat{y}_\ell)^2 = d_{k,\ell}^2, \; \ell \in \{1, 2, 3\}, \; k > 3$$

$$\Rightarrow \begin{bmatrix} 1 & -2\hat{x}_1 & -2\hat{y}_1 \\ 1 & -2\hat{x}_2 & -2\hat{y}_2 \\ 1 & -2\hat{x}_3 & -2\hat{y}_3 \end{bmatrix} \begin{bmatrix} x_\ell^2 + y_\ell^2 \\ x_\ell \\ y_\ell \end{bmatrix} = \begin{bmatrix} d_{1,k}^2 - \hat{x}_1^2 - \hat{y}_1^2 \\ d_{2,k}^2 - \hat{x}_2^2 - \hat{y}_2^2 \\ d_{3,k}^2 - \hat{x}_3^2 - \hat{y}_3^2 \end{bmatrix}$$

$$\Rightarrow z = (A^T A)^{-1} A^T b$$

$$\Rightarrow \hat{x}_k = [z]_2, \; \hat{y}_k = [z]_3$$

### B. Refining the coarse location estimates

After the ranging phase, every node broadcasts/receives the measured ranges to/from the rest nodes. For example, in Figure 5, node 1 will broadcast all the ranges involving itself i.e. $\{d_{1,2}, d_{1,3}, d_{1,4}\}$ and receive ranges of the other node pairs i.e. $\{d_{2,3}, d_{3,4}, d_{2,4}\}$. Available now on every node are all peer-to-peer ranging measurements. Solving the following non-linear least square problem results in all nodes’ location estimates.

$$\min_{p_i, p_j} \sum_{i < j} (\|p_i - p_j\| - d_{i,j})^2$$

s.t. $p_1 = [0, 0]^T$, $y_2 = 0$

Above minimization is addressed using Newton-Conjugate-Gradient (Newton-CG) method [3], in which the gradient and Hessian of the objective function w.r.t. nodes positions are calculated. Note that above procedure is carried out in parallel on every node.

Under the infrastructure-based scenario, especially in case of sparse anchor-deployment, we proposed a grid-based belief propagation method and its technical details may be found in [4].