



آزمایشگاه اینترنت اشیا

گزارش نهایی پروژه موقعیت یابی UWB

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال دوم ۰۴-۰۳

استاد:

جناب آقای دکتر اجلالی

اعضای گروه:

سینا ایمانی - ۹۹۱۰۰۳۱۴

حسین راعی - ۹۹۱۰۹۷۵۵

طاها اکبری - ۹۹۱۰۵۲۵۹

github.com/Taha1506/hardware-lab



فهرست مطالب

۲	۱ چکیده
۲	۲ مقدمه و انگیزه
۲	۱.۲ UWB و کاربرد آن در موقعیت‌یابی
۲	۲.۲ صورت مسئله پروژه
۲	۳ اهداف، دامنه و زمان‌بندی اولیه
۲	۴ طراحی سامانه (معماری کلان)
۲	۱.۴ اجزای اصلی
۳	۲.۴ تعاملات و لایه ارتباطی
۳	۵ روش‌ها و الگوریتم‌ها
۳	۱.۵ اندازه‌گیری فاصله با TWR
۳	۲.۵ برآورد موقعیت با کمینه‌سازی مربعات
۴	۳.۵ ملاحظات هندسی
۴	۶ پیاده‌سازی و مهندسی سیستم
۴	۱.۶ شبیه‌ساز Unity و آماده‌سازی
۴	۲.۶ ماژول‌های Tag و Anchor
۴	۳.۶ سرور برآورد موقعیت و LS
۵	۴.۶ پشتیبانی چند-تگی و فرمت خروجی
۵	۵.۶ ساختار پیام‌ها (نمونه)
۵	۷ آزمایش‌ها و ارزیابی
۵	۱.۷ سناریوهای آزمون
۵	۲.۷ شاخص‌های خطا
۵	۳.۷ جمع‌بندی نتایج کیفی
۶	۸ چالش‌ها و راهکارها
۶	۹ مستندسازی و رپوی کد
۶	۱۰ جمع‌بندی و پیشنهادها برای ادامه کار
۶	۱.۱۰ جمع‌بندی
۶	۲.۱۰ پیشنهادها فنی



۱ چکیده

در این پروژه، یک سامانه UWB برای **موقعیت‌یابی داخلی** مبتنی بر الگوریتم Two-Way Ranging (TWR) پیاده‌سازی و ارزیابی شد. معماری کلی شامل Tag‌های سیار، Anchor‌های ثابت با مختصات معلوم، و یک Server برای دریافت فاصله‌ها و انجام Multilateration (کمینه‌سازی حداقل مربعات) است. برای توسعه و آزمون سریع، یک محیط شبیه‌سازی در Unity ایجاد شد و پشته پیام‌رسانی بین اجزا طراحی گردید. در طول مسیر، چالش‌هایی مانند هم‌زمان‌سازی کلاک، پشتیبانی چندتگی، فرمت‌دهی پیام‌ها، و پایداری تخمین‌ها برطرف شدند و کدهای مربوطه در ریپوی پروژه منتشر گردید.

۲ مقدمه و انگیزه

۱.۲ UWB و کاربرد آن در موقعیت‌یابی

Ultra-Wideband (UWB) یک فناوری ارتباط بی‌سیم با پالس‌های بسیار کوتاه و پهنای باند وسیع است که به دلیل توان مصرفی پایین و دقت بالای اندازه‌گیری زمان پرواز (ToF)، برای ناوبری داخلی و رهگیری اشیاء مناسب است. با اندازه‌گیری ToF یا اختلاف زمان رسیدن (TDoA)، می‌توان فاصله میان گره‌ها را با دقت سانتی‌متری تخمین زد و سپس با روش‌های هندسی، مختصات برآوردی گره ناشناخته را محاسبه کرد.

۲.۲ صورت مسئله پروژه

هدف پروژه، ساخت یک مسیر کامل از کسب فاصله میان Tag و Anchor‌ها تا **تخمین موقعیت** در سمت سرور بود؛ به‌همراه ایجاد یک شبیه‌ساز برای تسریع توسعه و رفع وابستگی به سخت‌افزار. سند پروپوزال پروژه، چارچوب کلی (الگوریتم TWR، نقش اجزا، و زمان‌بندی) را تعیین کرده است.

۳ اهداف، دامنه و زمان‌بندی اولیه

- آشنایی عملی با UWB و ماژول‌هایی مانند DW1000 و مطالعه الگوریتم‌های موقعیت‌یابی.
- طراحی معماری Tag/Anchor/Server و تعریف فرمت پیام‌ها و قراردادهای شبکه.
- پیاده‌سازی الگوریتم TWR در سطح Tag/Anchor و Multilateration در سمت Server.
- توسعه محیط شبیه‌سازی در Unity برای تولید فاصله‌های شبه‌واقعی و تست چند-تگی.
- مستندسازی و یکپارچه‌سازی خروجی‌ها در ریپو. github.com/Taha1506/hardware-lab

۴ طراحی سامانه (معماری کلان)

۱.۴ اجزای اصلی

Tag: گره سیار که با ارسال/دریافت پیام‌های TWR فاصله‌اش تا هر Anchor را به‌دست می‌آورد.



Anchor: گره ثابت با مختصات معلوم که پیام‌های Ping/Response را مدیریت می‌کند و (در برخی نسخه‌ها) مختصات خود را نیز در پیام می‌گنجاند.

Server: ماژولی که فاصله‌های Anchor-Tag و مختصات Anchorها را دریافت و با کمینه‌سازی LS، موقعیت Tag را تخمین می‌زند.

۲.۴ تعاملات و لایه ارتباطی

ارتباط بین اجزا از طریق سوکت‌های TCP برقرار شده و پیام‌ها شامل نوع پیام، شناسه تگ/انکر، مهرزمان و در صورت لزوم مختصات Anchor هستند. در نسخه‌های بعدی، ساختار داده برای پشتیبانی چند تگ هم‌زمان بازطراحی شد تا داده‌ها بر اساس کلید «TagID» ذخیره و منتشر شوند؛ خروجی‌ها نیز شامل نام تگ، فاصله واقعی/تخمینی، و در نهایت «مختصات تخمینی» هستند.

۵ روش‌ها و الگوریتم‌ها

۱.۵ اندازه‌گیری فاصله با TWR

در Two-Way Ranging متقارن، دو تبادل پیام انجام می‌شود. اگر $t_{TX}^{(1)}$ و $t_{RX}^{(1)}$ به ترتیب زمان ارسال و دریافت در دور اول و $t_{TX}^{(2)}$ و $t_{RX}^{(2)}$ در دور دوم باشند، فرمول فاصله تقریبی به شکل زیر نوشته می‌شود (با فرض پاسخ تقریباً متقارن):

$$d \approx \frac{c}{4} [(t_{RX}^{(1)} - t_{TX}^{(1)}) - \Delta t_{reply}^{(1)} + (t_{RX}^{(2)} - t_{TX}^{(2)}) - \Delta t_{reply}^{(2)}], \quad (1)$$

که در آن c سرعت نور است و Δt_{reply} تأخیر ثابت/قابل کالیبره پاسخ است. در عمل، برای کاهش حساسیت به عدم هم‌زمانی کلاک، از نسخه‌های «دوطرفه متقارن» استفاده می‌شود و Δt ها در هر دور لحاظ و/یا کالیبره می‌شوند.

۲.۵ برآورد موقعیت با کمینه‌سازی مربعات

اگر $\mathbf{a}_i \in \mathbb{R}^n$ مختصات i Anchor و r_i فاصله اندازه‌گیری شده تا Tag باشد، مکان \mathbf{p} از حل کمینه‌سازی زیر به دست می‌آید:

$$\hat{\mathbf{p}} = \arg \min_{\mathbf{p}} \sum_{i=1}^m (\|\mathbf{p} - \mathbf{a}_i\|_2 - r_i)^2. \quad (2)$$

برای حل عددی، از خطی‌سازی حول تقریب اولیه و الگوریتم Gauss-Newton یا نسخه Weighted LS با وزن $w_i \propto 1/\sigma_i^2$ بهره می‌بریم. در پیاده‌سازی ما، یک حل Least Squares عملیاتی (با قابلیت جایگزینی با WLS) روی سرور اضافه شد.



۳.۵ ملاحظات هندسی

برای پایداری مسأله، آرایش Anchorها باید از هم خطی/هم صفحه‌ای پرهیز کند و تا حد امکان «زاویه‌های دید» متنوعی نسبت به Tag ایجاد نماید. همچنین، وجود حداقل $m \geq n + 1$ انکر ضروری است (با $n = 2$ یا 3).

۶ پیاده‌سازی و مهندسی سیستم

۱.۶ شبیه‌ساز Unity و آماده‌سازی

در ابتدای کار، ابزار Unity نصب و محیط آزمایش برای سنجش الگوریتم‌ها آماده شد. علاوه بر مطالعه روش‌های پیاده‌سازی، سازوکار تولید فاصله‌های شبه‌واقعی، مدیریت نویز، و ثبت خروجی‌ها طراحی گردید. نتیجه این فاز در رپوی پروژه موجود است.

۲.۶ ماژول‌های Tag و Anchor

در فاز میانی، کلاس‌های Tag و Anchor با منطق TWR تکمیل شدند. به دلیل عدم هم‌زمانی کلاک میان گره‌ها، Offsetهایی برای جبران تأخیرها و کالیبراسیون زمان پاسخ در نظر گرفته شد و فیلدهایی برای نوع پیام به ساختارها افزوده گردید.

شیوه پیدا کردن آیدی Tag و Anchor در دنیای واقعی. در پیام‌های ارسالی بین تگ و انکر، آیدی آن‌ها ارسال می‌شود تا مشخص باشد که طرفین پیام چه دستگاه‌هایی هستند. با توجه به مطالعات انجام‌شده، استاندارد جامعی برای پیدا کردن آیدی انکرها توسط تگ وجود ندارد و سازندگان مختلف روش‌های متفاوتی را به کار می‌برند. یکی از این روش‌ها، hard-code کردن آیدی انکرها درون هر تگ است. اما شیوه‌های دیگری نیز وجود دارد؛ برای مثال هر انکر می‌تواند به صورت تناوبی وجود خود را broadcast کند یا یک سرور مرکزی لیست انکرهای موجود را مخابره نماید. در پیاده‌سازی فعلی، لیست انکرها از خود بازی گرفته می‌شود که شبیه به حالت سرور مرکزی است.

۳.۶ سرور برآورد موقعیت و LS

در سمت سرور، کدی افزوده شد که با دریافت «مختصات انکرها» و «فاصله» Tag-Anchor مختصات Tag را با روش Least Squares محاسبه می‌کند. بدین منظور، پیام‌های ارسالی طوری تغییر یافتند که هر Anchor مختصات خود را نیز ارسال کند.

مدیریت آیدی دستگاه‌ها. همان‌طور که در بخش قبل اشاره شد، وجود آیدی یکتا برای تگ‌ها و انکرها ضروری است تا در سمت سرور داده‌ها به درستی تجمیع و پردازش شوند. در سناریوی فعلی، سرور فرض می‌کند که لیست انکرها از طریق بازی یا ماژول مرکزی در دسترس است و بر همین اساس پیام‌ها را تفسیر می‌کند. این رویکرد، ضمن سادگی، امکان توسعه بعدی برای استفاده از روش‌های استانداردتر در محیط واقعی را نیز فراهم می‌کند.



۴.۶ پشتیبانی چند-تگی و فرمت خروجی

ساختار داده‌ها برای چند تگ بازطراحی شد: به جای نگهداری حالت واحد، یک HashMap بر اساس TagID، اطلاعات مربوط به هر تگ (آخرین فاصله‌ها، زمان‌ها، تخمین موقعیت) را ذخیره می‌کند. خروجی نهایی شامل نام/شناسه تگ، فاصله واقعی و تخمینی، و مختصات برآوردی است؛ بعلاوه امکان ثبت و نمایش خروجی‌ها روی فایل فراهم شد تا بازپخش و ارزیابی ساده گردد.

۵.۶ ساختار پیام‌ها (نمونه)

• MSG_PING: { type, tag_id, ts_tx }

• MSG_RESP: { type, anchor_id, ts_rx, ts_tx, anchor_pos=(x,y[,z]) }

• MSG_RANGE: { type, tag_id, anchor_id, range, quality }

(توجه: فیلدها بسته به نسخه تغییر کرده‌اند؛ ایده اصلی حفظ اطلاعات زمان/پاسخ و ارسال مختصات انکر برای برآورد در سرور است.)

۷ آزمایش‌ها و ارزیابی

۱.۷ سناریوهای آزمون

۱. اعتبارسنجی تک‌تگ: جای‌گذاری 3 تا 4 انکر با آرایش پایدار و ارزیابی خطای مکان برای موقعیت‌های متنوع Tag.

۲. چندتگی هم‌زمان: سنجش رفتار سامانه با چند Tag فعال و بررسی تأثیر تناوب پیام‌ها بر پایداری LS.

۳. حساسیت به نویز/کالیبراسیون: تزریق نویز به ToF و بررسی اثر وزن‌دهی WLS.

۲.۷ شاخص‌های خطا

برای گزارش خطا از MAE و RMSE بین مختصات حقیقی و تخمینی استفاده می‌شود. همچنین «پراکندگی» تخمین‌ها در نقاط ثابت (با تکرار اندازه‌گیری) برای سنجش پایداری گزارش می‌گردد.

۳.۷ جمع‌بندی نتایج کیفی

- زنجیره پیام‌ها و مسیر محاسبه فاصله تا تخمین موقعیت به صورت **انتهای-به-انتهای** عمل می‌کند و در سناریوهای شبیه‌سازی شده خروجی پایدار تولید می‌شود.
- بازطراحی برای **چند تگ**، ثبت فایل خروجی و افزودن مختصات انکر در پیام‌ها، فرآیند ارزیابی را شفاف و قابل‌ردیابی کرده است.
- خطای مکانی تابع مستقیم کیفیت کالیبراسیون تأخیر پاسخ و هندسه انکرها است؛ استفاده از وزن‌دهی و طرح‌های کالیبراسیون بهتر، مسیر بهبود آینده است.



۸ چالش‌ها و راهکارها

عدم هم‌زمانی کلاک: با افزودن offset و لحاظ کردن تأخیر پاسخ در فرمول TWR جبران شد.

چندتگی: با بازطراحی ساختار داده‌ها مبتنی بر TagID و بازبینی فرمت پیام‌ها حل شد.

پایداری هندسی: توصیه به جانمایی مناسب انکرها (اجتناب از هم‌خطی/هم‌صفحه‌ای) و افزایش تعداد انکرها در محیط‌های پیچیده.

نویز و چندمسیره: پیش‌بینی استفاده از WLS، RANSAC و فیلترهای زمانی (مانند Kalman) برای حذف اندازه‌گیری‌های پرت.

۹ مستندسازی و ریپوی کد

تمام فازها (مطالعه اولیه، نصب، Unity پیاده‌سازی کلاس‌های Tag/Anchor، سرور LS، پشتیبانی Multi-Tag) به صورت مرحله‌ای در ریپو ثبت شده‌اند: github.com/Taha1506/hardware-lab.

۱۰ جمع‌بندی و پیشنهادها برای ادامه کار

۱.۱۰ جمع‌بندی

در این پروژه، یک زنجیره کامل برای موقعیت‌یابی داخلی مبتنی بر UWB از سطح تبادل پیام‌های TWR تا تخمین موقعیت در سمت سرور ساخته شد؛ شبیه‌ساز Unity فرآیند توسعه/آزمایش را تسهیل کرد و پشتیبانی چندتگی، سامانه را آماده سناریوهای واقعی‌تر نمود.

۲.۱۰ پیشنهادهای فنی

- گذار از LS به WLS با وزن‌دهی مبتنی بر کیفیت لینک و هندسه لحظه‌ای (DOP).
- فیلتر حالت (مانند EKF/UKF) برای هموارسازی مسیر تگ در طول زمان.
- کالیبراسیون سخت‌افزاری دقیق‌تر برای تأخیر پاسخ و جبران NLOS.
- افزودن لایه QoS و زمان‌بندی پیام‌ها برای مقیاس‌پذیری بهتر در چندتگی پرتراکم.