

АВТОНОМНА РОБОТИЗИРАНА СИСТЕМА ЗА СЛЕДЕНЕ НА ЛИНИЯ

Д. Славов

AUTONOMOUS ROBOTIC LINE TRACKING SYSTEM

D. Slavov

1000 София, бул. „Кл. Охридски“ 8, ТУ-София, бл. 9, катедра „Автоматизация на електропродвижванията“, каб.: 9406, тел: (02) 965-3945, d.slavov@tu-sofia.bg

Abstract.

This paper describes an autonomous mobile robot system for line tracking. It consists of a three-wheel platform, Arduino microcontroller and smart camera Pixy2 for computer vision. Based on the x and y coordinates at both ends of the visible line (vector), measured by the camera, vector gradient is calculated and weight factors are adaptively determined for generating a two-component error fed to a PID controller. The feedback control successfully drives the robot over the line even in the presence of sharp turns.

Key words: mobile robot, line tracking, computer vision, Arduino, Pixy

ВЪВЕДЕНИЕ

Следенето на линия е често използвано приложение в роботиката, защото предоставя основни функции за навигация, които се реализират сравнително лесно. В повечето роботи, които следят линия, се използват дискретни фотосензори, разграничаващи линията от фона. Този метод обаче е най-ефективен, когато се проследява по-широка линия, и освен това възприемането е локализирано в много тясна област, което затруднява или прави невъзможно да се предвидят промените в направлението на зададената траектория. В резултат на това при наличие на завой, по-остри от 90° , процесът по следене не винаги е успешен, като честотата за неуспех нараства с увеличаване на броя на тези завой и скоростта на мобилния робот.[1]

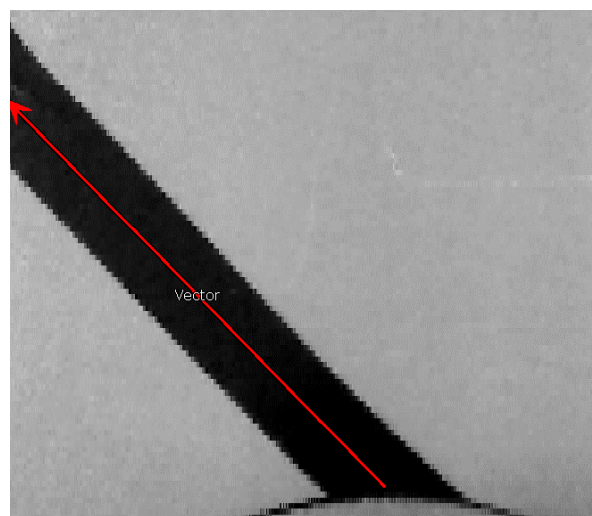
В настоящия доклад е представена система за автоматично управление на автономен мобилен робот при движението му върху предварително начертана линия. За измерване на положението спрямо линията се използва смарт камера Pixy2, която снима координатите на най-близката и най-отдалечената точка от текущо видимата част на линията. Тези данни се предават към микроконтролер Arduino, където се реализира управление въз основа на непосредственото и предстоящото състояние, както и степента на различие между двете.

ИЗЛОЖЕНИЕ

ПОЛУЧАВАНЕ НА ДАННИ ЗА СРЕДАТА

Смарт камерата Pixy2 обработва всеки кадър от заснетото видеосъдържание, намира линии, отговарящи на зададените критерии, и определя началото и края на всяка линия в кадъра. В общия случай интерес представлява само текущо следваната от робота линия. Ако в изображението се появят други, обикновено не е желателно роботът да прехвърли проследяването си върху тях. При обработването на кадрите софтуерът на камерата определя как се изменя всяка от намерените линии и запазва фокуса върху текущо проследяваната. Тя се обозначава като

вектор, чието начало се намира в най-долната (т.е. най-близката до робота) част от кадъра, в която се открива линия, и край съответно в най-горната такава част. Посоката на вектора – от началото към края – може да се използва за посока на придвижване.[3]



Фиг. 1. Изобразяване на вектор в софтуерното приложение PixyMon.

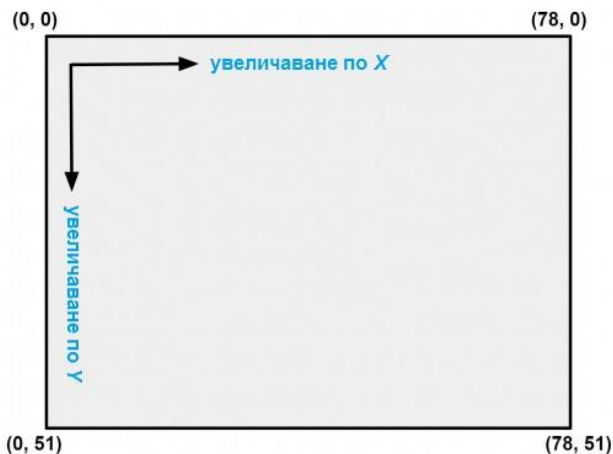
Алгоритъмът за откриване на линия намира съвкупност от гранични области (2 или 3 пиксела), чиято яркост силно се различава (съобразно зададената прагова стойност) от околните пиксели. Интересен е фактът, че в полученото първично изображение участват само пикселите с преобладаващо зелен цвят. Създава се вторично изображение с границите на линиите по всеки ред и колона. След това алгоритъмът сглобява достатъчно близките (отново съобразно зададена прагова стойност) граници в непрекъснати линии. От получените линии се избира онази, която е най-подходяща за целите на проследяването, и тя получава ролята на вектор. В последния етап се



Фиг. 3. Принципно схема на системата за управление

определя наличието на пресичания – техния общ брой, както и броя на разклонения във всяко пресичане.[4]

От всички данни, които камерата Pixu2 може да предава към микроконтролера, в докладваната тук разработката са използвани координатите по осите x и y на началната и крайната точки на вектора, т.е. общо 4 целочислени променливи. Тези координати се изразяват в пиксели от зрително поле с размер 79 x 52 пиксела.



Фиг. 2. Координатна система на зрителното поле

ОПИСАНИЕ НА РЕАЛИЗИРАНОТО АВТОМАТИЧНО УПРАВЛЕНИЕ

Процесът, чието управление описва настоящият доклад, се състои от придвижване на мобилна платформа върху тъмна линия на светъл фон. За тази цел по време на движение се компенсира грешка, представляваща отклонението спрямо средата на зрителното поле по хоризонтала. На всеки такт от работата на контролера се отчитат координатите на началото и края на вектора – изобразяващ отсечка, която попада в зрителното поле – и съобразно тях се изчислява отклонението спрямо центъра по оста x , който има стойност 40. Във формирането на грешката, която се подава към регулатора, участват 2 компоненти: отклоненията, измерени съответно в началото (e_{tail}) и в края (e_{head}) на вектора. Те получават различни теглови коефициенти, отразяващи различната степен на значимост, която оказват върху управлението на процеса. Увеличаването на теглото на e_{tail} води до по-точно следване на линията (платформата преминава с центъра си върху линията), а на e_{head} – до по-ранна реакция към предстоящото ѝ изменение.

Реализирано е автоматично изчисляване на тегловите коефициенти в хода на процеса, което адаптира влиянията на всяка от двете грешки към степента на промяна (градиента) на видимия участък от линията. За тази цел се

измерва ъгълът между вектора и мислената права, разполовяваща зрителното поле, т.е. правата, която свързва точките с координати (40, 0) и (40, 51), по формулата:

$$\alpha = \arctg\left(\frac{y_1 - y_0}{x_1 - x_0}\right) \cdot 57,2958 [^\circ] \quad (1)$$

където x_0 и y_0 са координатите на началото на вектора, а x_1 и y_1 – на края му.

Колкото по-голям е този ъгъл, толкова по-остър завои предстои и съответно толкова по-силно влияние трябва да има e_{head} , за да успее платформата навреме да завие.

В по-опростен вариант управлението може да се извършва само по началото на вектора, т.е. най-близката до робота точка от линията, но тогава няма да се получава информация за изменението на траекторията, по която роботът предстои да премине. В съществуващата към момента литература най-често са описани подобни системи, при които обаче насочването се извършва само по края на вектора, т.е. по най-отдалечената точка, която камерата вижда. При такава постановка мобилната платформа се стреми да застане централно върху все още недостигнат участък от линията и така се получава отклонение спрямо участъка, върху който в момента преминава – „скъсяване“ или „отрязване“ на завоите.

По формулата

$$e = e_{head} \cdot w_{head} + e_{tail} \cdot w_{tail} \quad (2)$$

където w_{head} и w_{tail} са съответните теглови коефициенти, се изчислява сумарната претеглена грешка, която се подава към софтуерно реализиран ПИД регулатор. Той изработва управляващо въздействие u чрез следната зависимост:

$$u = K_p e_n + K_i \sum_{x=0}^n e_x + K_d (e_n - e_{n-1}) \quad (3)$$

където e_n е грешката в сегашния такт, а e_{n-1} – в предишния. Коефициентът на интегриране K_i се умножава по сумата на всички досегашни грешки, която обаче има програмно ограничение и не може да нараства безкрайно.

Скоростта на въртене на двигателите се регулира чрез широчинно-импулсна модулация (ШИМ). Съответните изводи на микроконтролера подават разрешаващи сигнали за левия и десния двигател, като числовата стойност в диапазона [0, 255] съответства на широчината на импулса. Стойността u от (3) се добавя към максималната стойност $PIN_{max} = 255$ на извода, реализиращ ШИМ за левия двигател PIN_l , и се използва от тази за десния PIN_r .

Например при $u = -50$:

$$PIN_l = PIN_{max} + u = 255 - 50 = 205$$

$$PIN_r = PIN_{max} - u = 255 + 50 = 305 \text{ (ограничено до 255)}$$

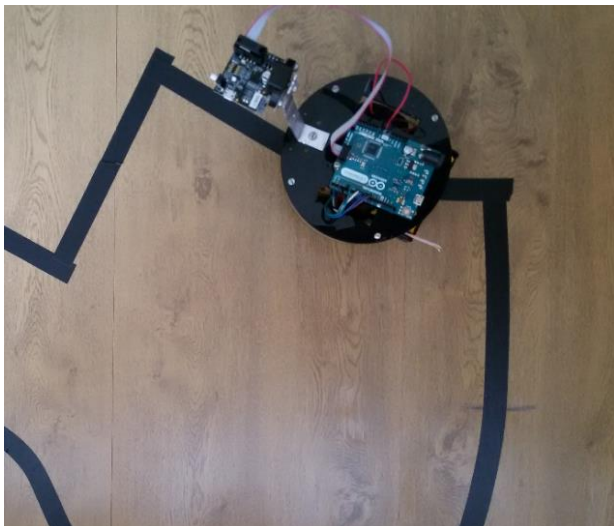
Тъй като $|PIN|$ не може да надвишава 255, а и поради преодоляване на загубите в използваните двигатели те не се завъртат при стойности на извода, по-малки от 40, то

действителната стойност на разрешаващия сигнал е ограничена в интервала $[-PIN_{max}, -40] \cup [40, PIN_{max}]$.

В зависимост от големината и знака на управляващото въздействие платформата завива по-рязко или по-плавно надясно или наляво и винаги поне единият от двигателите се върти с максимална скорост. В посочения пример платформата ще се движи напред и едновременно с това ще завива наляво.

АНАЛИЗ НА ЕКСПЕРИМЕНТИТЕ И РЕЗУЛТАТИТЕ

Проведените експерименти показват, че когато измененията в предвидената траектория са по-плавни (линията не прави завой под остър ъгъл), оптималните теглови коефициенти за e_{tail} и e_{head} са съответно около 95% и 5%. Това означава, че при линия с такъв характер дори малкото влияние на информираността за предстоящи завой осигурява правилно следене. Когато траекторията е с повече на брой и по-остри завой, тези коефициенти може дори да се изравнят – по 50%. Увеличаването на теглото за e_{tail} осигурява по-плавно придвижване на робота, но и увеличава вероятността той да пропусне остри завой в линията. Увеличаването на теглото за e_{head} придава по-голяма надеждност с оглед на това, че роботът по-трудно може да изгуби линията, но също така намалява точността, с която той преминава върху нея.



Фиг. 4. Изглед отгоре на роботизираната платформа в процес на следене

За оптимално протичане на управляемия процес са избрани експериментално следните коефициенти за ПИД регулатора: $K_p = 15$, $K_i = 1$, $K_d = 2$.

Пропорционалната съставка е с максималната стойност, позволяваща достатъчно плавно придвижване. При силно променлива траектория може да се извлече полза от диференциална компонента, която е значително по-голяма ($K_d \approx K_p$) от избраната, но това също така води до ненужни резки движения на робота при наличие на различни смущения в средата – напр. по-различна осветеност в даден участък, отблясъци, прекъсване или нагъване на материала, с който е изобразена линията.

Ако действието на интегралната част е твърде агресивно, системата за автоматично управление може да стане неустойчива. Регулаторът би могъл да предизвика пререгулиране на грешката и дори да създаде нова грешка с по-голяма амплитуда и обратен знак. Когато това се случи, регулаторът ще започне непрекъснато да променя

изхода си между минималната и максималната стойност – т.нар. явление „гонитба“.[2] В разглежданата система „гонитбата“ се изразява в поведение, при което мобилната платформа почти не се придвижва напред по линията, а се върти силно наляво и надясно. Управляваният процес – който включва не само промяна на скоростта на двигателите, а цялостното придържане на платформата върху следената линия – е много по-бавен от регулатора, чието бързодействие зависи от времето за едно изпълнение на програмата (15 – 17 ms, т.е. тактова честота около 60 Hz). По тази причина би се акумулирала голяма грешка за времето, докато центърът на платформата се върне върху линията. Ето защо е избран коефициент на интегриране K_i с много малка стойност – около 15 пъти по-малък от коефициента на пропорционалност K_p . Освен това акумулираната в интеграла грешка се нулира, когато бъде компенсирала текущата грешка. С други думи, интегралната съставка действа само от момента на поява на отклонение спрямо линията до момента, в който това отклонение изчезне. Това помага за избягване на пререгулиране, каквото в конкретната постановка е нежелателно.

Реализираният ПИД закон придава по-голяма прецизност в управлението на процеса. За разлика например от платформи, които използват няколко инфрачервени сензора и са удобни за следене на линия с малък брой скорости на завиване във всяка посока, тук диапазонът на измервани стойности е много по-голям, което позволява по-плавно регулиране.

ЗАКЛЮЧЕНИЕ

В настоящия доклад е разглеждана мобилна роботизирана платформа, която автономно се придвижва по предварително начертана линия. С помощта на смарт камера за техническо зрение PiXu 2 се получават параметрите, описващи средата – координати на началото и края на участък от линията (наричан „вектор“), който попада в зрителното поле на камерата. Тези параметри се предават към микроконтролер Arduino, който посредством двигателите на платформата реализира автоматично управление на процеса по следене. Разработена е софтуерна програма, която изчислява два компонента на отклонението спрямо заданието – центъра на зрителното поле. Координатите на вектора служат също за определяне на градиента на линията, съобразно който динамично се изчисляват теглови коефициенти за всеки от двата компонента. Формираната по този начин грешка се подава на входа на ПИД регулатор, който реализира управляващо въздействие към двигателите.

Благодарение на регулирането по ПИД закон, изпреварващата информация за предстоящата траектория и динамично определяната степен на влияние, което измерваните параметри оказват върху поведението, системата успешно се придвижва както в слабо променлива среда, така и по траектория с близко разположени остри завой. При често използваните постановки с точкови фотосензори липсват някои от горепоспаните възможности за получаване на данни, което на свой ред ограничава подходите към управлението. Представеното тук решение може да се използва като тяхна алтернатива или като основа за надграждане с допълнителни сензори и оптимизиране на управляващите алгоритми с тяхно участие.

Интерес за бъдещи разработки представлява възможността да се добави функция за определяне на разликата в яркостта между линията и фона, което ще позволи на

системата да следи светла линия върху тъмен фон. Това може да става в началото и дори динамично в хода на процеса, тъй като приложният програмен интерфейс на камерата осигурява лесно превключване между двата режима посредством член-функцията `setMode()`. За по-голяма независимост от околното осветление системата може да бъде подпомагана от интегрираните в камерата светодиоди, но поради отразяването на светлината им от по-гладки или гланцирани повърхности това може да внесе допълнителни смущения във възприемането на средата, така че са необходими допълнителни експерименти и анализи.

ЛИТЕРАТУРА

1. Христов В., Дурчев С., „Следене на траектория от мобилен робот zumo“, Годишник на ТУ-София, 2018, том 68, книга 3, 113-120стр., ISSN 1311-0829.
2. ПИД регулиране, сп. Инженеринг ревю – брой 9, 2006 г. (<https://www.engineering-review.bg/bg/pid-regulirane/2/552/> – достъпен на 20.08.2019 г.).
3. Документация за камерата Pixy 2, раздели „Line tracking for line-following“ и „Line tracking quickstart“ (<https://docs.pixycam.com/wiki/doku.php?id=wiki:v2:start> – достъпен на 20.08.2019 г.).
4. Форум за потребители и разработчици на продуктите Pixy, раздели „Pixy Software“ и „Pixy Hardware“ (<https://discourse.pixycam.com/> – достъпен на 20.08.2019 г.).
5. Изходен код на вградените програми в Pixy (https://github.com/charmedlabs/pixy2/blob/master/src/device/main_m4/src/line.cpp – достъпен на 20.08.2019 г.) – при разработване на софтуера.