Управление на мобилен робот посредством смарт камера за машинно зрение

Д. Славов

Mobile Robot Control Using Machine Vision Smart Camera

D. Slavov

Key Words: *Mobile robot; automatic control; computer vision; machine vision; Arduino; Pixy; line tracking.*

Abstract. *This paper presents ways to control a mobile robot system for two different tasks: line tracking; and finding and passing through an opening in a vertical plane. The system is constructed of a three-wheel platform, Arduino microcontroller and a smart camera Pixy2 for machine vision. For the line tracking task, based on the x and y coordinates at both ends of the visible line, 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. For the hole-passing task, the robot locates a color-illuminated opening in a vertical plane, moves frontally and passes through it. The system finds the frontal position by repeatedly measuring the opening width while maneuvering straight and diagonally towards it. Both tasks are successfully implemented despite the variable ambient lightning conditions.*

Увод

Мобилните роботи представляват комплексни системи, които могат да съдържат множество различни сензори и изпълнителни механизми и се характеризират най-вече с програмируемо управление [1] и адаптивност към динамично променяща се работна среда.

В статията се разглеждат методи за автономно придвижване на мобилна роботизирана платформа, която с помощта на смарт камера за машинно зрение изпълнява задачи за придвижване по линия и за преминаване през отвор във вертикална равнина. Съществуват редица изследвания, свързани с приложението на този тип камери в комбинация с нискобюджетен микроконтролер като Arduino или Raspberry. Например за следене на линия и разпознаване на обекти с приложение при автономно управляемите, следване на инвалидна количка от колесна платформа за пренасяне на багаж и много други разработки, използващи функциите на разглежданата тук смарт камера за разпознаване на отделни и комбинации от хомогенни зони със зададен цвят. 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 завои, по-остри от 90° процесът по следене невинаги е успешен, като честотата за неуспех нараства с увеличаване на броя на тези завои и скоростта на мобилния робот [1].

В представената система е разработено автоматично управление на мобилен робот при движението му върху предварително начертана линия, както и при намиране и преминаване през отвор във вертикална равнина. За измерване на положението спрямо линията се използва смарт камера Pixy2, която снема 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. В задачата за преминаване през отвор мобилният робот търси и намира оптимална по даден критерий позиция спрямо определена цел. Това се осъществява чрез установяване на промяната в ширината на целта с течение на времето. В системата не се използват никакви други сензори като например за измерване на разстояние. Поради това тя е с ниска себестойност и опростена структура, но също така може да служи като базов модел за по-нататъшно надграждане и прецизиране на изпълняваните функции.

Използвани компоненти

За целите на това изследване са използвани следните компоненти: микроконтролер Arduino Leonardo, кръгло шаси, към което са монтирани постояннотокови двигатели и монтирани към тях колела, интегрална схема L298N, сервоплатформа, камера Pixy2.

Смарт камера за машинно зрение. CMUcam е устройство за компютърно зрение от нисък ценови клас, предназначено за изследвания в областта на роботиката.

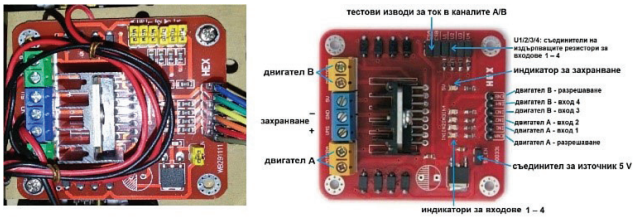
Състои се от малка видеокамера и микроконтролер със сериен интерфейс. Докато много от другите цифрови камери обикновено използват връзка със значително по-широка честотна лента, опростеният интерфейс на CMUcam позволява по-лесна работа с микроконтролери. Освен това вграденият микропроцесор осигурява първично обработване на изображенията и проследяване на хомогенни зони със зададен цвят. Устройството също така има изключително малък формфактор, което допринася за пригодността му при направата на малки мобилни роботи. Първоначално се разработва от Университет „Карнеги Мелън“ (Carnegie Mellon University), а впоследствие лицензът се предоставя на различни производители [2].

Петото поколение на изделията CMUcam (т.е. CMUcam5) е получило името Pixy, а използваното тук устройство е от версия 2, наречена Pixy2. В тази система видеопреобразовател заснема изображения с кадрова честота 60 Hz, които се предават към вградения микропроцесор, обработват се и резултатите се изпращат към свързания микроконтролер. Това може да става по различни комуникационни интерфейси: UART, I2C, USB[3], както и използвания в разглежданата система SPI през порта ICSP (три канала, реализирани чрез изводите 1, 3 и 4 на Arduino).

За откриване на обекти се използва алгоритъм, наречен цветово свързани компоненти (Color Connected Components, CCC), който реализира метода на етикетиране на свързаните компоненти (Connected Components Labeling) в цветни изображения.

Смарт камерата Pixy2 разполага с шест (2 групи по 3) извода, чрез които може да управлява самостоятелно до два серводвигателя. Тази функция е използвана при построяването на платформата, като така се оползотворят по-пълно възможностите на устройството и се опростява както схемата на свързване, така и софтуерната програма.

L298N. За превключване на постояннотоковите двигатели се използва електронна схема L298N (модел WB291111), реализираща двоен H-мост. Този драйвер може да приема сигнали за транзисторно-транзисторна логика (ТТЛ), чрез които да управлява индуктивни товари като релета, соленоиди, постояннотокови и стъпкови двигатели. Окомплектован е с два входа, които независимо разрешават или забраняват работата на всеки от двата двигателя. Има и допълнителен захранващ вход, така че логическото управление да се изпълнява при по-ниско напрежение.



Фиг. 1. Външен вид и означения на изводите в L298Np

Скоростта на въртене на двигателите се регулира чрез широчинно-импулсна модулация (ШИМ). Изводите 9 и 10 на Arduino подават разрешаващи сигнали към входовете ENB и ENA съответно за левия и десния двигател. Числовата стойност в диапазона [0, 255] съответства на продължителността на импулса $t_{импулс}$ чрез следното отношение:

$$(1) \qquad \frac{t_{импулс}}{T} = \frac{PIN}{PIN_{макс}}$$

където *PIN* и $PIN_{макс}$ са съответно текущата и максималната (255) стойност, подавана на извода. Периодът е определен от тактовата честота, с която работят съответните изводи: в случая 490 Hz, т. е. T ≈ 2 ms. Ако подадената стойност на извода е например 100, тогава

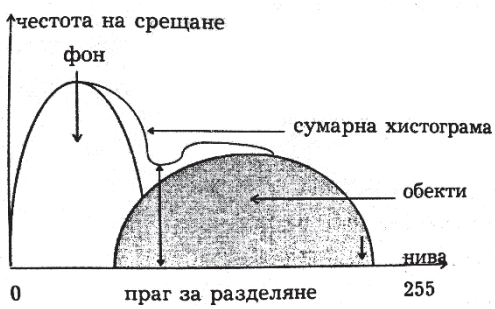
$$t_{импулс} = \frac{100 \,.\, 2}{255} \approx 0{,}8\ ms$$

и съответно $t_{пауза} = T - t_{импулс} \approx 1{,}2$ ms.

На практика поради преодоляване на загубите в използваните двигатели (те не се завъртат при стойности на извода, по-малки от 40) минималният коефициент на запълване D е около 16%.

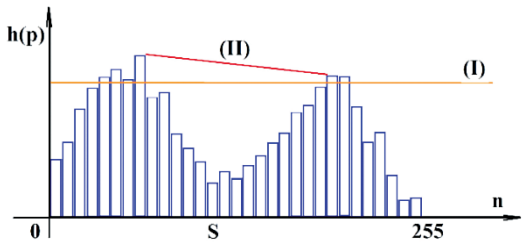
Някои използвани в PIXY2 методи за обработка на изображения

Прагово детектиране на зоните на обектите. Това е разпространен подход, който се използва успешно на етапа за предварителна обработка за получаване на единичните зони на обектите, обхванати от нулевата зона на фона. Бинарното преобразуване се осъществява по информация в хистограмата от нива на сиво на входното изображение. Ако това изображение съдържа група обекти със сходни физически параметри, разположени върху контрастен фон, зоните на обектите и фона са хомогенни и добре различими. Хистограмата има два обособени локални максимума и изразена долина между тях.



Фиг. 2. Хистограма на нивата на сиво в контрастно изображение

При тези условия се прилагат алгоритми за сегментация, основани на детектиране на зоните на обекти с глобален праг. В първия етап от изследването на хистограмата се определя разположението на двата максимума: пресича се обвивката ѝ с прагова стойност, която нараства до откриване на нивата, в които сеченията на двата максимума се свиват до сегменти с единична ширина. Във втория етап двата максимума се съединяват с праволинеен сегмент и се търси най-голямото разстояние от него до обвивката – то съответства на прага за бинаризация [4].



Фиг. 3. Избор на праг за бинаризация

Построяване на скелетон. Преобразуването на зоната на отделен обект в средни оси я трансформира в линейно описание, наречено скелетон. То се получава с алгоритми за изтъняване на двумерните зони, като се запазва свързаността и информативните линейни размери. Алгоритмите се основават на формализъм за определяне на разстоянието *r* от произволен пиксел P до линия от пиксели A

$$r(P, A) = MIN \{R(P, Z); Z \square A\},$$

където *R* е евклидов критерий за разстоянието между точка *P* и точките *Z* от линията A [4].

Сегментиране на хомогенни зони с оператори за съседство и качество на пикселите. Хомогенните зони представляват области от изображението, съставени от съседни пиксели с еднакво качество (цвят, градация). В двуградационните изображения качеството се определя еднозначно от стойността на пикселите 0 или 1, а при изображения с нива на сиво то представлява множество от непресичащи се диапазони от градации, определени чрез анализ на хистограмата. За еднозначно определяне на съседство може да се използва оператор за 6-съседство [4].

Преобразуване RGB – HSV. Филтриращите методи въз основа на цвета са широко използвани, тъй като са бързи, ефикасни и сравнително устойчиви. Изображението се конвертира от цветовия модел RGB (red, green, blue – червено, зелено, синьо) към HSV (hue, saturation, value – цвят, наситеност, стойност). По същество пространството HSV представлява проекция на цветовия куб RGB върху нелинеен хроматичен ъгъл, процент на радиално насищане (сатурация) и стойност на яркостта. Това декомпозиране има за цел да раздели данните за яркостта от тези за цвета и да приближи изображението към по-лесни за възприемане цветови модели. Например ако даден алгоритъм за компютърно зрение трябва да въздейства само върху стойността (яркостта) на изображението, но не върху насищането или цвета му, едно просто решение би било използването на цветовите отношения

$$(2)\quad r = \frac{R}{R+G+B},\ g = \frac{G}{R+G+B},\ b = \frac{B}{R+G+B}.$$

След манипулиране на яркостта, например посредством изравняване на хистограми, всеки цвят може да се умножи с отношението на новата спрямо старата яркост, за да се получи коригирана тройка от червено, зелено и синьо [5].

Изчислените стойности за цвета и наситеността на всеки пиксел, получен от преобразователя, се използват като параметри на филтриране. Това цели запазване на постоянен цвят на обекта дори при различно ниво на осветеност и експозиция [3].

Действието на камерата Pixy 2 при изпълнение на програмата за цветово свързани компоненти използва цвета на обектите вместо тяхната яркост. Той се формира в две измерения U и V:

$$(3)\quad U = \frac{R-G}{R+G+B},\quad V = \frac{B-G}{R+G+B},$$

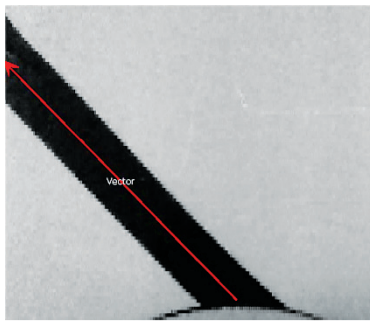
където R – червен, G – зелен, B – син цвят. След това се определят границите на *U* и *V*, с които е свързан даден обект. Тъй като при този метод се дели на сумата от всички цветни пиксели, яркостта в голяма степен се пренебрегва и това намалява влиянието на промените в осветеността.

Етикетиране на свързаните компоненти. При този метод изображението се сканира и пикселите в него се групират в компоненти съобразно свързаността си, т.е. всички пиксели в един свързан компонент имат близък по стойност интензитет и по някакъв начин са свързани помежду си. След определяне на групите всеки пиксел получава етикет, представляващ ниво на сиво или цвят (цветово етикетиране) съгласно компонента, към който принадлежи [6].

Задача 1. Следене на линия

Получаване на данни за средата

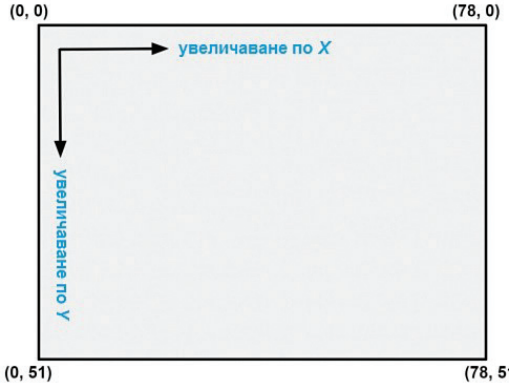
Смарт камерата Pixy2 обработва всеки кадър от заснетото видеосъдържание, намира линии, отговарящи на зададените критерии, и определя началото и края на всяка линия в кадъра. В общия случай интерес представлява само текущо следваната от робота линия. Ако в изображението се появят други, обикновено не 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. 4. Изобразяване на вектор в софтуерното приложение PixyMon

Алгоритъмът за откриване на линия намира съвкупност от гранични области (2 или 3 пиксела), чиято яркост се различава силно (съобразно зададената прагова стойност) от околните пиксели. 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 всички данни, които камерата Pixy2 може да предава към микроконтролера, в представената тук разработка са използвани координатите по осите x и y на началната и крайната точки на вектора, т.е. общо четири целочислени променливи. Тези координати се изразяват в пиксели от зрително поле с размер 79 x 52 пиксела.



Фиг. 5. Координатна система на зрителното поле

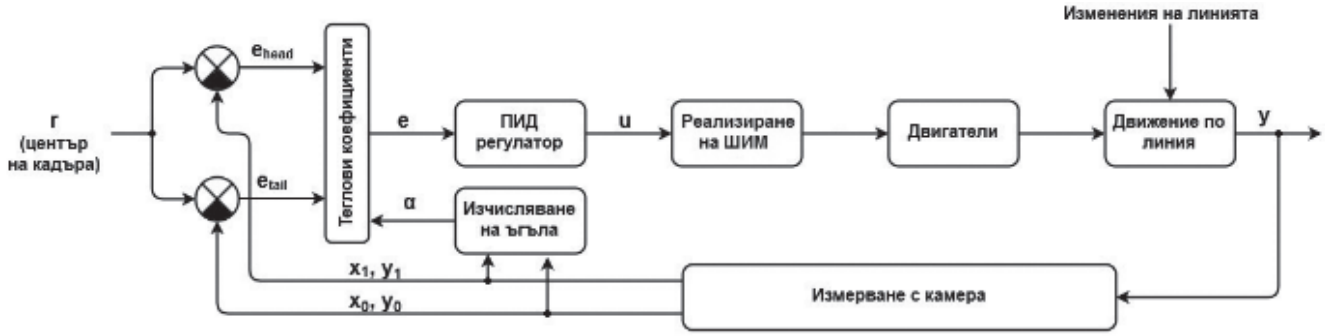
Описание на реализираното автоматично управление

Процесът, чието управление е описано в тази статия, 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 две 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

$$(1) \qquad \alpha = arctg(\frac{y_1 - y_0}{x_1 - x_0}).57{,}2958\ [^\circ]\ ,$$

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



Фиг. 6. Принципна схема на системата за управление

Колкото по-голям е този ъгъл, толкова по-остър завой 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

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

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:

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

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 изважда от тази за десния PIN_r.

Например при u = –50:

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

$PIN_r = PIN_{max} - u = 255 + 50 = 305$ (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, 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.

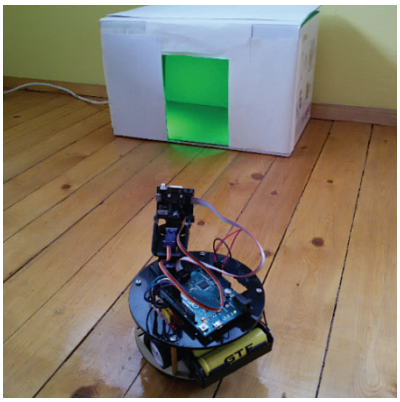
Задача 2. Намиране на отвор и преминаване през него

Настройване на камерата

Конфигурирането на цветни сигнатури се извършва чрез софтуера PixyMon. В него може да се определят и запишат до седем различни цвята и когато камерата установи участък от зрителното си поле, отговарящ на вече конфигуриран цвят, тя го разпознава като хомогенна зона. В термините на PixyMon е възприето тези зони да се наричат *блокове* и независимо от формата им те биват апроксимирани до правоъгълници със страни, успоредни на координатните оси на кадъра. Текущо откритите хомогенни зони се записват в масив от обекти, всеки от които съдържа следните данни: пореден номер на сигнатурата (т.е. на уникалния цвят, зададен на зоната), координати по x и по y (изразени като пиксели по ширина и височина на зрителното поле) на центъра на зоната, ширина и височина (отново в пиксели), индекс и брой поредни кадри (от 0 до 255), в които хомогенната зона е присъствала в зрителното поле.

Опитна постановка

Използвана е кутия във формата на правоъгълен паралелепипед с изрязан в нея правоъгълен отвор, който е с приблизително същата ширина като тази на мобилната платформа. Височината е достатъчна, за да може платформата да премине, но конкретната ѝ стойност няма значение, тъй като не участва в изчисляването на траекторията.



Фиг. 7. Външен вид на мобилната платформа и отвора

Във вътрешността на кутията е монтирано осветление със зелен цвят и достатъчна яркост. Кутията е поставена върху равната повърхност, по която се придвижва мобилната платформа.

Алгоритъм

Чрез разработения метод мобилният робот сам установява местоположението на отвора, заема фронтална спрямо него позиция и приближавайки се, преминава през него. За да може да премине през отвора, роботът най-напред трябва да застане фронтално, защото в противен случай ширината на отвора няма да е достатъчна. Търсенето на фронтална позиция (ФП) се осъществява чрез многократна последователност от: 1) придвижване под ъгъл диагонално спрямо целта и 2) придвижване директно към целта. Първата част от тази последователност осигурява приближаване към ФП, а втората компенсира ситуации, в които роботът едновременно се отдалечава от ФП и се приближава към целта – в такъв случай ширината на отвора няма да намалява и отдалечаването от ФП няма как да бъде установено. Критерий за доближаване до ФП е отчетеното от камерата увеличаване на ширината на отвора по време на диагоналното придвижване: ширината се измерва веднъж преди началото на движението и втори път след края му, като двете стойности се сравняват. Последователността 1 – 2 се изпълнява, докато е налице приближаване към ФП. Тъй като интерес представлява само изменението на ширината – а не точната ѝ стойност – алгоритъмът е приложим в широки граници на отдалеченост от ФП и от отвора.

Фиг. 8. Блокова схема на алгоритъма за намиране на отвор и фронтална позиция

Първоначално мобилният робот започва да търси отвор (цел), като се завърта надясно около мислената ос в центъра на шасито си. Щом камерата открие участък, осветен в предварително зададения цвят, значи целта е намерена, тъй като този участък на практика изобразява отвора в кутията. Следва ориентиране на шасито с предната си част към целта, измерване и регистриране на ширината на отвора. Роботът се завърта под определен ъгъл (в разработката е използван приблизително 45°) и се придвижва напред за определено време (като независимо от това камерата продължава да следи целта). Отново се измерва ширината и двете се сравняват. Ако по-късно измерената ширина е по-голяма, значи е налице приближаване към целта, приближаване към ФП или и двете. Последователността от движение по диагонал и директно към целта се повтаря, докато в даден момент (след първото измерване) се окаже, че по-късно измерената ширина е по-малка от по-рано измерената. Това означава, че роботът току-що е подминал ФП и е започнал да се отдалечава от нея. В такъв случай се извършва придвижване назад за определено време, което действие приблизително съответства на връщане във ФП. От тази позиция роботът се насочва към отвора и започва да се движи напред, докато премине през него.

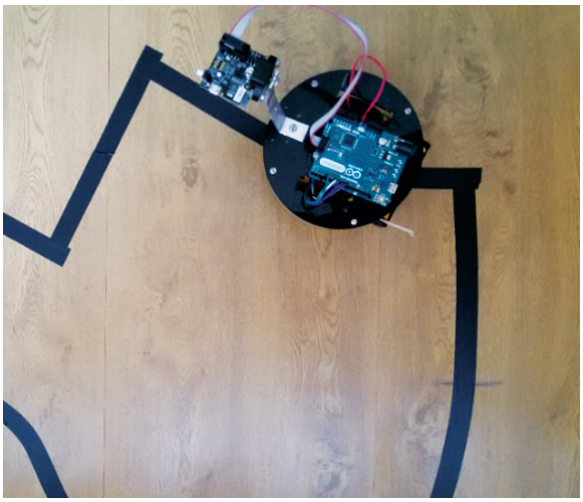
Ако след първото диагонално придвижване се установи, че роботът се е отдалечил от целта, това означава, че първоначално избраният диагонал е неподходящ и трябва да се избере обратният. Затова роботът, без да променя ориентацията си, се връща назад за определено време, ориентира се към целта (т.е. застава в изходно положение) и отново се завърта под същия ъгъл, но в противоположна посока – която ще се използва и до края на програмата.

Описание на извършените експерименти и анализ на резултатите

Следене на линия

Проведените експерименти показват, че когато измененията в предвидената траектория са по-плавни (линията не прави завои 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 завои осигурява правилно следене. Когато траекторията е с повече на брой и по-остри завои, тези коефициенти могат 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 завои 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.

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



Фиг. 9. Изглед отгоре на роботизираната платформа в процес на следене

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 – например 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* [7]. 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 компенсирана 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.

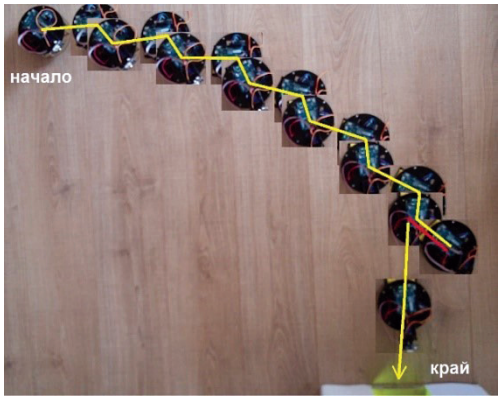
Намиране и преминаване през отвор

Мобилната система е изпробвана при няколко начални положения, различаващи се по отдалечеността си спрямо целта и ФП. По време на експериментите не беше достигнато максималното разстояние, до което камерата открива хомогенната зона от цветни пиксели и следователно не възникнаха свързани с това затруднения.

Когато началното положение е твърде далече от целта, не е обосновано веднага да започне търсене на ФП, защото то ще бъде по-неточно, отколкото от близко разстояние и освен това ако роботът застане във ФП много далече от отвора, по време на финалното си придвижване има по-силни предпоставки правилната му траектория да се наруши – например поради неравности по пътя и нееднакви (въпреки заданието) скорости на левия и десния двигател. Тези смущения имат по-слабо влияние, когато финалното придвижване започва близо до целта.

В зависимост от разстоянието до целта и до ФП, както и от ъгъла, който шасито сключва с оста на ФП по време на диагоналното движение, са възможни ситуации, при които роботизираната платформа едновременно се приближава към целта и се отдалечава от ФП. Когато увеличаването на видимата ширина вследствие на приближаването към целта е по-голямо или равно на намаляването ѝ вследствие на отдалечаването от ФП, измерената ширина няма да намалява и роботът ще продължи да се движи в неправилна посока. За компенсиране на тези ситуации в алгоритъма е въведена стъпката за движение директно към целта, която служи и за промяна на ъгъла, сключен между шасито и оста на ФП.

На *фиг. 10* е показана траекторията (в жълто), по която платформата се придвижва към целта си. С червено е означено връщането след подминаване на отчетената ФП.



Фиг. 10. Траектория на движението

При програмното реализиране на алгоритъма са отчетени ограниченията, свързани с размера на зрителното поле на камерата Pixy2 – 316 x 208 пиксела. В случаите, когато роботизираната система завива, ако това става прекалено бързо или движението на сервоплатформата е прекалено бавно, камерата ще загуби следения обект. За намаляване на тези ситуации е съобразена скоростта на завиване, както и бързодействието на сервоуправлението, реализирано софтуерно като ПД звено.

Използването на самостоятелен светлинен източник, благодарение на който от отвора се излъчва светлина с постоянен интензитет, позволява на системата да работи в широки граници на изменение на околното осветление. Затруднения възникват основно при наличие на друг светлинен източник (например силна дневна светлина от прозорец), разположен зад обекта, както и на гладки повърхности, в които достатъчно силно се отразява светлината от обекта и следователно такива постановки следва да се избягват.

За експериментите е избрано осветяване на отвора със зелен цвят, тъй като наличието на този цвят в околните обекти на конкретната опитна постановка е най-малко. Така системата в голяма степен се предпазва от откриване на нежелана цел в началното положение или евентуално след изгубване на планираната цел. Също така осветеността на отвора се регулира чрез прегради, поставени в кутията, за да се постигнат граници и размер на регистрирания от камерата обект, които най-точно съответстват на действителния отвор.

Заключение

Разгледани са решения за автономно придвижване на мобилна роботизирана платформа по предварително начертана линия и за преминаване през отвор във вертикална равнина. С помощта на смарт камера за машинно зрение Pixy2 се получават параметрите, описващи средата – координати на началото и края на участък от линията или координати на центъра и ширина на хомогенна зона със зададен цвят. Тези параметри се предават към микроконтролер Arduino, който посредством двигателите на платформата реализира автоматично управление на процесите.

Изграждащите компоненти са нискобюджетни, което прави системата икономически ефективна. Самата смарт камера е с много малки размери: 43 x 39 x 16 mm и тегло около 10 g. Това позволява да бъде окомплектована в ограничени пространства, например като вграден елемент от компактен мобилен робот. Тъй като не се разчита на персонален компютър (в ролята на хост) за обработване на изображенията, а този процес е интегриран в самото устройство, отпада необходимостта от предаване на данните към допълнителен обработващ модул и оттам – към изпълнителния механизъм.

При следене на линия се установяват координатите на вектора, служещи за определяне на градиента на линията, съобразно който динамично се изчисляват теглови коефициенти за всеки от двата компонента на грешката. Така формирана, тя 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 оп-

ределяната степен на влияние, което измерваните параметри оказват върху поведението, системата се придвижва успешно както в слабо променлива среда, така и по траектория с близко разположени остри завои. При често използваните постановки с точкови фотосензори липсват някои от описаните по-горе 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.

Проведените експерименти показват, че системата може да се използва успешно за насочване на мобилен робот към отвор във вертикална равнина, например в стена или между други обекти. Тя би могла да намери приложение като част от насочването на транспортни роботи в среда с пространствени ограничения, автономно паркиране на електрически превозни средства и други. Основен принос на разглежданото решение е възможността за получаване на представа за размера на дадена цел без използване на допълнителни сензори за разстояние, а единствено чрез геометрични пресмятания. Освен това в конкретната разработка целта представлява отвор, а не твърдо тяло и това би затруднило работата и/или влошило точността на често използваните сензори като например инфрачервени и ултразвукови.

Една от потенциалните възможности за бъдещо развитие на системата е включването на още сензори: например разположени на допълнителни ротационни платформи, за да се измерва най-близкото разстояние до равнината (например стена), което на свой ред да помогне за оптимизиране на траекторията на придвижване. Също така в алгоритъма може да се включи информация за положението (или няколко поредни във времето положения) на сервомеханизма, носещ камерата Pixy2, и така да се получава по-добра представа за ориентацията на мобилната платформа.

Литература

1. Христов, В., С. Дурчев. Следене на траектория от мобилен робот zumo. Годишник на ТУ – София, 68, 2018, книга 3, 113-120, ISSN 1311-0829.

2. CMUcam. https://en.wikipedia.org/wiki/CMUcam.

3. Документация на продуктите Pixy. https://docs.pixycam.com.

4. Венков, П. Анализ и разпознаване на изображения и сцени. Технически университет – София, 1996.

5. Richard Szeliski. Computer Vision: Algorithms and Applications, 3 September 2010 draft © 2010 Springer.

6. Fisher, R., S. Perkins, A. Walker and E. Wolfart. Connected Components Labeling. 2003. https://homepages.inf.ed.ac.uk/rbf/HIPR2/label.htm.

7. ПИД регулиране. – *Инженеринг ревю*, 2006, № 9 (https://www.engineering-review.bg/bg/pid-regulirane/2/552/).

8. Изходен код на вградените програми в Pixy (https://github.com/charmedlabs/pixy2/blob/master/src/device/main_m4/src/line.cpp) – при разработване на софтуера.

** Адресите в интернет са посетени на 22.08.2019 г.*

За контакти:
Ас. **Данаил Славов**
Катедра „Автоматизация
на електрозадвижванията"
Технически университет – София
e-mail: d.slavov@tu-sofia.bg