

Univerza v Ljubljani
Fakulteta za elektrotehniko

Andrej Zdešar

**VODENJE AVTONOMNEGA MOBILNEGA
SISTEMA Z ROBOTSKO ROKO NA OSNOVI
DIGITALNE SLIKE**

DOKTORSKA DISERTACIJA

Ljubljana, 2015

Univerza v Ljubljani
Fakulteta za elektrotehniko

Andrej Zdešar

**VODENJE AVTONOMNEGA MOBILNEGA
SISTEMA Z ROBOTSKO ROKO NA OSNOVI
DIGITALNE SLIKE**

DOKTORSKA DISERTACIJA

Mentor: prof. dr. Igor Škrjanc, univ. dipl. inž. el.
Somentor: izr. prof. dr. Gregor Klančar, univ. dipl. inž. el.

Ljubljana, 2015

Zahvala

Ob tej priliki naj se zahvalim vsem, ki so ne le pripomogli k nastanku tega dela, temveč tudi vsak po svoje doprinesli k nekaj prekrasnim trenutkom v mojem življenju.

Zahvaljujem se mentorju prof. dr. Igorju Škrjancu, univ. dipl. inž. el., in somentorju izr. prof. dr. Gregorju Klančarju, univ. dipl. inž. el., za vse strokovne nasvete, ideje, spodbudo in pomoč pri študiju in raziskovalnem delu. Posebna zahvala gre mentorju, ki mi je omogočil sodelovanje z različnimi zunanjimi raziskovalnimi skupinami.

Zahvala gre vsem sodelavcem v Laboratoriju za modeliranje, simulacijo in vodenje in Laboratoriju za avtonomne mobilne sisteme na Fakulteti za elektrotehniko Univerze v Ljubljani za prijetno delovno vzdušje, koristne nasvete in inovativne ideje.

Zahvaljujem se tudi prof. dr. Jorgeju Pomaresu in njegovim sodelavcem, da so me sprejeli kot gostujočega raziskovalca v svojem laboratoriju na Univerzi v Alicantu.

Hvala staršem, sestri in ostalim sorodnikom za vso podporo, trud in razumevanje.

Zahvala gre vsem prijateljem in kolegom, ki so mi vedno stali ob strani.

Iskreno hvala.

Ljubljana, 7. 4. 2015

Andrej Zdešar

Povzetek

Delo obravnava načrtovanje in izvedbo vodenja mobilnega sistema s pristopom vodenja na osnovi slike (VOS). V ospredju obravnave je izdelava mobilnega sistema, ki lahko avtonomno opravi podano nalogo, pri čemer uporablja kamero kot osnovni senzor za zaznavanje okolja. V ta namen je bilo razvitih več podpornih sistemov — le-ti so predmet obravnave v doktorski disertaciji —, kot so: sistem za vodenje po trajektoriji na osnovi slike, sistem za načrtovanje poti na sliki in sistem za ocenjevanje stanj oz. lokalizacijo. Sistemi so bili izdelani predvsem za vodenje kolesnih mobilnih sistemov po trajektoriji, ki je eden izmed temeljnih problemov pri vodenju mobilnih sistemov. Določene uspešne izvedbe sistemov iz vodenja kolesnih mobilnih sistemov smo zasnovali tudi za vodenje robotske roke na osnovi slike. Pri izvedbi algoritmov vodenja smo se omejili na uporabo le ene kamere, pri čemer so bile naš predmet zanimanja predvsem nekanonične postavitve kamere v okolju. Sistem vodenja smo zasnovali tako, da lahko nalogo podamo neposredno na sliki. Zato smo regulirni zakon želeli izvesti po metodi direktnega vodenja na osnovi slike, kar pomeni, da je pogrešek v regulator definiran neposredno na sliki. Za problem sledenja trajektoriji smo predstavili izvedbo dvoprostostnega regulatorja s prediktivno regulacijo na osnovi modela sistema v povratni zanki. Prikazali smo postopek načrtovanja in možnosti optimizacije referenčne trajektorije z zlepkami Bernstein-Bézierjevih parametričnih krivulj. Pri načrtovanju poti smo posebej obravnavali določene posebne primere, ko mobilni sistem doseže ničelno tangencialno hitrost. Ker v takšnih primerih model sistema za izvedbo regulatorja postane nevodljiv, smo vpeljali model pomožnega pogreška, ki omogoča stabilno rešitev problema, čeprav smo pri tem morali sprejeti določene kompromise. Pri načrtovanju regulatorja za vodenje po trajektoriji se pojavi problem določevanja lege mobilnega sistema v okolju, zato smo se posvetili tudi problemu izvedbe lokalizacije z uporabo slikovne informacije. Za potrebe izračuna

pogreška in ocenjevanja lege smo uporabili opis slik z redkimi lokalnimi značilkami, ki omogočajo strnjen in učinkovit zapis slikovne vsebine. V delu je prikazana tudi možnost izvedbe kalibracije sistema VOS s kamero na osnovi gibanja, ki ne zahteva posebnega prirejanja okolja le za namen kalibracije. Veliko predlaganih pristopov temelji na obravnavi in uporabi posebnih lastnosti perspektivne projekcije ravnine. Predlagane rešitve smo eksperimentalno preizkusili tako v simulacijskem okolju kot tudi na realnih sistemih.

Abstract

This work presents the use of visual servoing approaches for solving the control problems in mobile robotics. The main emphasis is on development of a mobile robotic system that is able to autonomously complete the desired task based only on visual information provided by camera mounted on the mobile system. In order to achieve this goal, several support systems were developed — these systems are presented in this work —, like systems for: image-based trajectory tracking, reference trajectory generation, mobile robot localization. One of the main focus of the work is on development of trajectory tracking control of wheeled mobile robots. The presented control approach was also used in development of trajectory tracking control of robotic arm. The setup consists of a single camera that is mounted on a mobile system in some arbitrary non-canonical way. The presented control design enables in-image definition of the task. An image-based visual servoing approach is used to solve the control problem. A 2 DOF internal-model control approach with a predictive control algorithm in the feedback loop is used to solve the trajectory tracking problem. Bernstein-Bézier parametric splines are used for trajectory generation. The properties of Bernstein-Bézier curves at boundary points were studied for the special case when tangential velocity reaches zero. Since the model of a system at zero tangential velocity becomes uncontrollable, a special model of the error is introduced that enables for stable control solution. The control error is obtained using local image features and methods from optimal state estimation. An approach for camera calibration is presented that does not require special preparation of the environment for the purpose of calibration and it is based on rotating object motion. Many of the presented approaches were designed based on some special properties of perspective transformation of a plane. The presented methods were experimentally evaluated in simulation environment and on real systems.

Vsebina

Seznam slik	xiii
Seznam tabel	xxi
Seznam simbolov	xxiii
Simbolni in znakovni zapis	xxv
1 Uvod	1
2 Pregled področja	5
3 Zaznavanje s kamero	27
3.1 Perspektivna projekcija	27
3.2 Geometrija več pogledov	32
3.3 Kalibracija kamere	33
3.3.1 Kalibracija na podlagi parov točk	34
3.3.2 Kalibracija s pomočjo stožnic	35
3.4 Kalibracija iz gibanja	37
3.4.1 Direktna metoda: Ocena homografije na podlagi parov točk	38
3.4.2 Lourakisova metoda: Ocena homografije na podlagi koplarnih krogov	39
3.4.3 Predlagana metoda: Ocena homografije iz gibanja	40
3.4.4 Rezultati	44
3.4.5 Diskusija	49

4	Strojna obdelava slik	53
4.1	Nizkonivojska obdelava slik	53
4.2	Metode obdelave slik z lokalnimi značilkami	58
4.2.1	Detekcija lokalnih tipičnih vzorcev	58
4.2.2	Opis z značilkami	62
4.2.3	Iskanje parov	65
4.3	Sledenje vzorcev skozi sekvenco slik	66
4.4	Ocena premika iz sledenja značilk	71
5	Vodenje mobilnega sistema po poti	75
5.1	Kinematični model mobilnega sistema	75
5.2	Ocenjevanje stanj	77
5.2.1	Ocenjevanje lege na podlagi vizualne in kolesne odometrije	78
5.2.2	Opazovalnik stanj	79
5.3	Vodenje po trajektoriji	84
5.4	Prediktivni regulator na osnovi modela pogreška	86
5.4.1	Model pogreška	88
5.4.2	Model pomožnega pogreška	99
6	Načrtovanje poti	113
6.1	Parametrične krivulje	113
6.1.1	Optimizacija trajektorije	114
6.2	Bernstein-Bézierjeve krivulje	115
6.2.1	Posebne lastnosti v 2D	117
6.2.2	Zlepki	132
6.2.3	Optimizacija poti	134
6.2.4	Perspektivična transformacija	144
6.2.5	Postopek načrtovanja poti	145
7	Vodenje robotske roke	149
7.1	Načrtovanje trajektorije	149
7.2	Regulator za sledenje trajektoriji	152
8	Sklep	157

9	Doprinos k znanosti	159
	Dodatek	161
A	Preslikave med koordinatnimi sistemi	163
B	Geometrijski model kamere	165
C	Diskretna linearna transformacija (DLT) za ravnino	171
D	Središčni razteg	175
	Literatura	177

Seznam slik

2.1	Naslovnice nekaterih pomembnejših knjig s področja strojnega vida, avtonomnih mobilnih sistemov, robotike in vodenja na osnovi slike.	6
2.2	Zgradba vodenja na osnovi slike ¹ .	8
2.3	Osnovni prostori.	10
2.4	Konceptualni prikaz različnih oblik VOS. Na slikah (a) in (b) je LVOS, na slikah (c) in (d) je PVOS, na slikah (e) in (f) je DVOS. Levi stolpec prikazuje kaskadne izvedbe VOS s pomožno regulacijsko zanko, medtem ko desni stolpec prikazuje neposredne izvedbe. $\mathbf{R}$ je model robotskega sistema, $\mathbf{C}$ je model kamere, $\mathbf{G}$ in $\mathbf{G}_p$ je sta glavni in pomožni regulator, $\mathbf{R}^{-1}$ in $\mathbf{C}^{-1}$ sta inverzna modela robotskega sistema in kamere; $\mathbf{s}_r$ je vektor želenih vrednosti značilnk, $\mathbf{s}$ je vektor trenutnih vrednosti značilnk, $\mathbf{y}$ je vektor meritev na sliki, $\mathbf{u}$ je regulirni signal, $\mathbf{e}$ je regulacijski pogrešek; $\mathbf{w}_p$, $\mathbf{y}_p$ in $\mathbf{e}_p$ so signali pomožnega regulatorja.	12
2.5	Razdelitev VOS glede na način obdelave slike.	17
3.1	(a) <i>Camera obscura</i> v človeški velikosti postavljena v pristanišču mesta Cape Town in (b) posnetek slike, ki je nastala na stenah znotraj naprave.	28
3.2	Vplivi treh osnovnih preslikav na obliko kvadrata, od leve proti desni: podobnostna, afina in čista perspektivična preslikava.	30
3.3	Preslikava krožnice in vzporenih premic na ravnini (levo) preko perspektivične transformacije na slikovno ravnino (desno).	36
3.4	Dve različni hipotezi o sliki središča kroga (svetovna ravnina na levi in slikovna ravnina na desni) — $\Delta^{\{P,W\}}\varphi_i = \{P,W\}\varphi_{i+1} - \{P,W\}\varphi_i$.	42

3.5	Meritve položajev točk na sliki pri različnih deležih obsega celega kroga, ki ga opravi točka v ravnini kroženja, od (a) do (f) so deleži sledeči: 0,4, 0,5, 0,6, 0,7, 0,8 in 0,9.	46
3.6	Analiza kvalitete ocene pri različnih deležih celega kroga (slika 3.5) za vse tri obravnavane metode z različnimi merili: (a) merilo d , (b) merilo D in (c) merilo RAE	46
3.7	Izmerjeni položaji točk na sliki z različnimi variancami normalno porazdeljenega šuma s srednjo vrednostjo nič, od (a) do (f) so variance šuma sledeče: 1 px^2 , 2 px^2 , 4 px^2 , 8 px^2 , 16 px^2 in 32 px^2 . .	46
3.8	Analiza kvalitete ocene pri različnih variancah šuma pri merjenju položajev točk na sliki (slika 3.7) za vse tri obravnavane metode z enakimi merili kot na sliki 3.6.	47
3.9	Izmerjeni časi z različnimi vrednostmi uniformo porazdeljenega šuma s srednjo vrednostjo nič, od (a) do (f) so vrednosti šuma sledeče: 0 s, 0,01 s, 0,02 s, 0,03 s, 0,05 s in 0,08 s.	47
3.10	Analiza kvalitete ocene pri različnih vrednostih šuma pri merjenju časa (slika 3.9) za vse tri obravnavane metode z enakimi merili kot na sliki 3.6.	47
3.11	Izmerjeni položaji točk na sliki pri različnih naklonih slikovne ravnine glede na ravnino kroženja, od (a) do (f) so nakloni sledeči: 0° , 10° , 20° , 50° , 60° in 70°	48
3.12	Analiza kvalitete ocene pri različnih naklonih (slika 3.11) za vse tri obravnavane metode z enakimi merili kot na sliki 3.6.	48
3.13	Realni podatki za oceno homografije med ravnino vrtenja (gibanja mobilnega sistema) in slikovno ravnino.	50
3.14	Preslikava ravnine pogleda 3.13 na virtualno ravnino vrtenja, ki je ocenjena z različnimi metodami: (a) z direktno metodo, (b) z Lourakisovo metodo, (c) s predlagano metodo in (d) z direktno metodo ob upoštevanju kotne hitrosti ocenjene s predlagano metodo.	50
4.1	(a) Postopno glajenje slike z Gaussovim filtrom in (b) Gaussova piramida glajenih in decimiranih slik.	60
4.2	Shema za sledenje tipičnih vzorcev skozi sekvenco slik.	68

4.3	Detektirane tipične točke z algoritmom SIFT pri dveh premaknjenih postavitev mobilnega sistema s kamero.	69
4.4	(a) Najdeni pari med značilkami na slikah 4.3a in 4.3b z metodo primerjanja vektorjev značilk in (b) robustno ocenjeni pari točk z algoritmom RANSAC.	70
4.5	Preslikava slike 4.3b na sliko 4.3a.	70
4.6	Koordinatni sistemi (vozlišča) in preslikave med njimi (povezave).	72
5.1	Model dvokolesnega mobilnega sistema na diferencialni pogon.	76
5.2	Odpertozančno vodenje pri neničelnem začetnem pogrešku ($e_x(0) = 0,5$ m, $e_y(0) = 0,5$ m in $e_\varphi(0) = 0,5$ rad). (a) Referenčna (krepko vijolično) in dejanska trajektorija ter (b) krmilna signala.	85
5.3	Lega referenčnega mobilnega sistema glede na mobilni sistem.	89
5.4	Sledenje trajektoriji s prediktivnim regulatorjem pri ničelnem začetnem pogrešku in normalno porazdeljenemu belemu šumu s standardno deviacijo $\sigma_{n,x} = \sigma_{n,y} = 0,002$ m in $\sigma_{n,\varphi} = 0,002$ rad na izhodu. (a) Referenčna (krepko) in dejanska trajektorija, (b) pogrešek, (c) regulirna signala in (d) lege mobilnega sistema vsakih polnih pet sekund.	93
5.5	Sledenje trajektoriji s prediktivnim regulatorjem pri neničelnem začetnem pogrešku ($e_x(0) = 0,5$ m, $e_y(0) = 0,5$ m in $e_\varphi(0) = 0,5$ rad). Opisi slik (a) do (d) so podani v opisu na sliki 5.4.	94
5.6	Sledenje trajektoriji s prediktivnim regulatorjem pri neničelnem začetnem pogrešku ($e_x(0) = 0,5$ m, $e_y(0) = 0,5$ m in $e_\varphi(0) = 0,5$ rad) in normalno porazdeljenemu belemu šumu s standardno deviacijo $\sigma_{n,x} = \sigma_{n,y} = 0,002$ m in $\sigma_{n,\varphi} = 0,002$ rad na izhodu. Opisi slik (a) do (d) so podani v opisu na sliki 5.4.	95
5.7	Sledenje trajektoriji s prediktivnim regulatorjem pri neničelnem začetnem pogrešku ($e_x(0) = 0,5$ m, $e_y(0) = 0,5$ m in $e_\varphi(0) = 0,5$ rad) in normalno porazdeljenemu belemu šumu s standardno deviacijo $\sigma_{n,x} = \sigma_{n,y} = 0,002$ m in $\sigma_{n,\varphi} = 0,002$ rad na izhodu. V trenutku $t = 60$ s nastopi na izhodu motnja, ki izmakne mobilni sistem iz svoje lege. Opisi slik (a) do (d) so podani v opisu na sliki 5.4. . . .	96

5.8	Sledenje trajektoriji s prediktivnim regulatorjem na realnem mobilnem sistemu. Opisi slik (a) do (d) so podani v opisu na sliki 5.4.	98
5.9	Lega referenčnega mobilnega sistema glede na mobilni sistem s pomožno točko.	99
5.10	Sledenje trajektoriji s pomožnim regulatorjem (5.64) pri neničelnem začetnem pogrešku ($e_x(0) = 0,5$ m, $e_y(0) = 0,5$ m in $e_\varphi(0) = 0,5$ rad). (a) Referenčna in pomožna referenčna trajektorija, (b) sledenje pomožne točke in mobilnega sistema, (c) sledenje pomožne točke pomožni referenčni trajektoriji, (d) sledenje mobilnega sistema referenčni trajektoriji, (e) lege mobilnega sistema ob vsaki polni sekundi in (f) regulirna signala.	103
5.11	Pogrešek pri sledenju trajektoriji s pomožnim regulatorjem (5.64). (a) Časovni potek pogreška mobilnega sistema in (b) časovni potek pomožnega pogreška.	104
5.12	Sledenje trajektoriji s prediktivnim regulatorjem z modelom pomožnega pogreška pri ničelnem začetnem pogrešku in normalno porazdeljenemu belemu šumu s standardno deviacijo $\sigma_{n,x} = \sigma_{n,y} = 0,002$ m in $\sigma_{n,\varphi} = 0,002$ rad na izhodu. (a) Referenčna (krepko) in dejanska trajektorija, (a) referenčna pomožna (krepko) in dejanska pomožna trajektorija, (c) pogrešek, (c) pomožni pogrešek, (e) regulirna signala in (f) lege mobilnega sistema vsakih polnih pet sekund.	105
5.13	Sledenje trajektoriji s prediktivnim regulatorjem pri neničelnem začetnem pogrešku ($e_x(0) = 0,5$ m, $e_y(0) = 0,5$ m in $e_\varphi(0) = 0,5$ rad). Opisi slik (a) do (f) so podani v opisu na sliki 5.12.	106
5.14	Sledenje trajektoriji s prediktivnim regulatorjem pri neničelnem začetnem pogrešku ($e_x(0) = 0,5$ m, $e_y(0) = 0,5$ m in $e_\varphi(0) = 0,5$ rad) in normalno porazdeljenemu belemu šumu s standardno deviacijo $\sigma_{n,x} = \sigma_{n,y} = 0,002$ m in $\sigma_{n,\varphi} = 0,002$ rad na izhodu. Opisi slik (a) do (f) so podani v opisu na sliki 5.12.	107

5.15	Sledenje trajektoriji s prediktivnim regulatorjem pri neničelnem začetnem pogrešku ($e_x(0) = 0,5\text{ m}$, $e_y(0) = 0,5\text{ m}$ in $e_\varphi(0) = 0,5\text{ rad}$) in normalno porazdeljenemu belemu šumu s standardno deviacijo $\sigma_{n,x} = \sigma_{n,y} = 0,002\text{ m}$ in $\sigma_{n,\varphi} = 0,002\text{ rad}$ na izhodu. V trenutku $t = 60\text{ s}$ nastopi na izhodu motnja, ki izmakne mobilni sistem iz svoje lege. Opisi slik (a) do (f) so podani v opisu na sliki 5.12.	108
5.16	Sledenje trajektoriji s prediktivnim regulatorjem z modelom pomožnega pogreška na realnem mobilnem sistemu. Opisi slik (a) do (f) so podani v opisu na sliki 5.12.	110
6.1	Primer BB-krivulje s kontrolnimi točkami.	116
6.2	(a) Vpliv kontrolnih točk $\mathbf{p}_0$, $\mathbf{p}_1$ in $\mathbf{p}_2$ na lastnosti BB-krivulje v začetni robni točki pri predpostavki, da je začetna tangencialna hitrost različna od nič ($v_0 \neq 0$). Časovni profil (b) tangencialne hitrosti in (c) kotne hitrosti.	122
6.3	(a) Posebna postavitev kontrolne točke $\mathbf{p}_2 = 2\mathbf{p}_1 - \mathbf{p}_0$, ki pri predpostavki o neničelni začetni hitrosti $v_0 \neq 0$ oz. $\mathbf{p}_0 \neq \mathbf{p}_1$, zagotovi, da sta tako tangencialni pospešek (vodoraven začetek hitrostnega profila (b)) kot (c) kotna hitrost v začetni robni točki enaka nič.	123
6.4	(a) Vpliv kontrolnih točk $\mathbf{p}_0$, $\mathbf{p}_2$ in $\mathbf{p}_3$ na lastnosti BB-krivulje v začetni robni točki pri predpostavki $v_0 = 0 \wedge a_0 \neq 0$. Časovni profil (b) tangencialne hitrosti in (c) kotne hitrosti.	126
6.5	(a) Posebna postavitev kontrolne točke $\mathbf{p}_3 = 3\mathbf{p}_2 - 2\mathbf{p}_0$, ki pri predpostavki $v_0 = 0 \wedge a_0 \neq 0$ oz. $\mathbf{p}_0 = \mathbf{p}_1 \neq \mathbf{p}_2$, zagotovi, da sta tako prvi odvod tangencialnega pospeška (vpliva na obliko tangencialne hitrosti (b)) kot (c) kotna hitrost v začetni robni točki enaka nič.	127
6.6	(a) Vpliv kontrolnih točk $\mathbf{p}_0$, $\mathbf{p}_3$ in $\mathbf{p}_4$ na lastnosti BB-krivulje v začetni robni točki pri predpostavki $v_0 = 0 \wedge a_0 = 0 \wedge \dot{a}_0 \neq 0$. Časovni profil (b) tangencialne hitrosti in (c) kotne hitrosti.	130

6.7	(a) Posebna postavitev kontrolne točke $\mathbf{p}_4 = 4\mathbf{p}_3 - 3\mathbf{p}_0$, ki pri predpostavki $v_0 = 0 \wedge a_0 = 0 \wedge \dot{a}_0 \neq 0$ oz. $\mathbf{p}_0 = \mathbf{p}_1 = \mathbf{p}_2 \neq \mathbf{p}_3$, zagotovi, da sta tako drugi odvod tangencialnega pospeška (vpliva na obliko tangencialne hitrosti (b)) kot (c) kotna hitrost v začetni robni točki enaka nič.	131
6.8	Najkrajša pot pri podanih zahtevah v robnih točkah in podanim časom trajanja trajektorije (optimizacija s prosto vmesno točko). .	137
6.9	Optimalna kratka pot pri podanih zahtevah v robnih točkah in podanim časom trajanja trajektorije (optimizacija s prosto vmesno točko)..	139
6.10	Optimalna maksimalna hitrost pri podanih zahtevah v robnih točkah in podanim časom trajanja trajektorije (optimizacija s prosto vmesno točko)..	140
6.11	Optimalna maksimalna kotna hitrost pri podanih zahtevah v robnih točkah in podanim časom trajanja trajektorije (optimizacija s prosto vmesno točko)..	140
6.12	Optimalna maksimalna kotna hitrost pri podanih zahtevah v robnih točkah in podanim časom trajanja trajektorije (optimizacija s prosto vmesno točko).	141
6.13	Optimalna ukrivljenost pri podanih zahtevah v robnih točkah in podanim časom trajanja trajektorije (optimizacija s prosto vmesno točko).	141
6.14	Optimalna ukrivljenost pri podanih zahtevah v robnih točkah in podanim časom trajanja trajektorije (optimizacija s prosto vmesno točko).	142
7.1	Primer referenčne trajektorije kamere v (a) 3D-prostoru in prikaz trajektorij štirih referenčnih točk na sliki.	151
7.2	Vodenje kamere po trajektoriji s prediktivnim regulatorjem (sledenje štirim točkam). Referenčna in dejanska trajektorija v (a) 3D-prostoru kamere in (b) na sliki ter (c) poteki tangencialnih in kotnih hitrosti kamere.	155
B.1	Predstavitev lege kamere in slike v prostoru.	166

B.2 Model tanke leče.	166
-------------------------------	-----

Seznam tabel

3.1	Primerjava različnih metod ocenjevanja homografije.	43
3.2	Primerjava ocenjenih slik središča kroga z različnimi metodami na realnih podatkih za štiri različne postavitve mobilnega sistema. . .	50
5.1	Cenilke pri sledenju trajektoriji s prediktivnim regulatorjem za (a) primer s slike 5.4, (b) primer s slike 5.5, (c) primer s slike 5.6 in (d) primer s slike 5.7.	97
5.2	Cenilke pri sledenju trajektoriji s prediktivnim regulatorjem na realnem mobilnem sistemu.	98
5.3	Cenilke pri sledenju trajektoriji s prediktivnim regulatorjem z modelom pomožnega pogreška za (a) primer s slike 5.12, (b) primer s slike 5.13, (c) primer s slike 5.14 in (d) primer s slike 5.15.	109
5.4	Cenilke pri sledenju trajektoriji s prediktivnim regulatorjem z modelom pomožnega pogreška na realnem mobilnem sistemu.	111

Seznam simbolov

Simbol	Opis
$\boldsymbol{v}$	Stolpčni vektor.
$\boldsymbol{M}$	Matrika.
$\boldsymbol{M}^T$	Transponirana matrika.
$\boldsymbol{M}^{-1}$	Inverzna matrika.
$\boldsymbol{M}^{-T}$	Transponirana inverzna matrika.
$\boldsymbol{M}^\dagger$	Psevdoinverzna matrika.
${}^F\boldsymbol{p}$	Točka podana v koordinatnem sistemu F .
${}^F\tilde{\boldsymbol{p}}$	Točka v koordinatnem sistemu F podana v homogenih koordinatah.
${}^Fx, {}^Fy, {}^Fz$	Elementi vektorja ${}^F\boldsymbol{p}$.
${}^F\boldsymbol{q}$	Posplošene koordinate podane v koordinatnem sistemu F .
${}^F\boldsymbol{t}_E$	Premik med izhodiščema koordinatnih sistemov F in E .
${}^Ft_{E,x}, {}^Ft_{E,y}, {}^Ft_{E,z}$	Elementi vektorja ${}^F\boldsymbol{t}_E$.
${}^F\boldsymbol{R}_E$	Rotacijska matrika med koordinatnima sistemoma F in E .
${}^Fr_{E,i,j}$	Element v i -ti vrstici in j -tem stolpcu matrike ${}^F\boldsymbol{R}_E$.
${}^F\boldsymbol{T}_E$	Homogena transformacijska matrika med koordinatnima sistemoma F in E .
${}^Ft_{E,i,j}$	Element v i -ti vrstici in j -tem stolpcu matrike ${}^F\boldsymbol{T}_E$.
$\mathring{q} = (q_0, \boldsymbol{q}) = (q_0, q_x, q_y, q_z)$	Kvaternion.

Simbol	Opis
$\mathring{q}(\mathbf{v}) = (0, \mathbf{v})$	Vektor v zapisan v obliki kvaterniona.
$\mathring{\mathbf{q}} = [q_0, q_x, q_y, q_z]^T$	Kvaternion zapisan v obliki stolpčnega vektorja.
${}^E \mathring{q}_F$	Kvaternion, ki predstavlja rotacijo med koordinatnima sistemoma F in E .
$\mathbf{v}_1^T \mathbf{v}_2, \mathbf{v}_1 \cdot \mathbf{v}_2$	Skalarni produkt.
$[\mathbf{v}_1]_{\times} \mathbf{v}_2, \mathbf{v}_1 \times \mathbf{v}_2$	Vektorski produkt.
$\mathbb{P}^n$	n -dimenzionalni projekcijski prostor.
$\mathbb{R}^n$	n -dimenzionalni realni prostor.
$\mathbb{C}^n$	n -dimenzionalni kompleksni prostor.
$\mathbb{Z}^n$	n -dimenzionalni prostor celih števil.
$\mathbb{N}^n$	n -dimenzionalni prostor naravnih števil.
$\mathbb{H}$	Prostor kvaternionov.
$O(n)$	Ortogonalni prostor.
$SO(n)$	Posebni ortogonalni prostor (prostor rotacij).
$E(n)$	Evklidski prostor.
$SE(n)$	Posebni Evklidski prostor (prostor homogenih transformacij).
s	Laplaceov operator.
z	z operator.
T	Čas vzorčenja.
v	Tangencialna hitrost.
ω	Kotna hitrost.
x, y	Koordinate na 2D-sliki.
X, Y, Z	Koordinate v 3D-prostoru.
$\text{Rot}_x(\varphi),$	Rotacija okoli osi x za kot φ .
$\text{Rot}_y(\varphi),$	Rotacija okoli osi y za kot φ .
$\text{Rot}_z(\varphi),$	Rotacija okoli osi z za kot φ .
$\text{Rot}(\mathbf{a}, \varphi),$	Rotacija okoli osi $\mathbf{a}$ za kot φ .
ϕ_R, φ	Kot nagiba.
ϕ_P, θ	Kot naklona.
ϕ_Y, ψ	Kot odklona.

Simbolni in znakovni zapis

Simbol	Oznaka	Opis
${}^A\mathbf{p} \in \mathbb{R}^3$	pA	Krajevni vektor izražen v koordinatnem sistemu A .
${}^A\mathbf{t}_{C,B} \in \mathbb{R}^3$	tC1B2A	Vektor iz C v B izražen v koordinatnem sistemu A .
${}^A\mathbf{t}_B \in \mathbb{R}^3$	tB2A	Vektor iz A v B izražen v koordinatnem sistemu A .
${}^A\mathbf{R}_B \in \text{SO}(3)$	rB2A	Rotacijska matrika, ki podaja orientacijo koordinatnega sistema B glede na koordinatni sistem A .
${}^B\mathring{q}_A \in \mathbb{H}$	qB2A	Kvaternion, ki podaja orientacijo koordinatnega sistema B glede na koordinatni sistem A in ustreza rotacijski matriki ${}^A\mathbf{R}_B$.

Simbolni zapis	Znakovni zapis
${}^A\mathbf{R}_B = {}^A\mathbf{R}_C {}^C\mathbf{R}_B$	rB2A = rC2A*rB2C
${}^B\mathring{q}_A = {}^B\mathring{q}_C {}^C\mathring{q}_A$	qB2A = qB2C*qC2A
${}^A\mathbf{p} = {}^A\mathbf{R}_B {}^B\mathbf{p} + {}^A\mathbf{t}_B$	pA = rB2A*pB + tB2A
$\mathring{q}({}^A\mathbf{p}) = {}^B\mathring{q}_A \mathring{q}({}^B\mathbf{p}) {}^B\mathring{q}_A^{-1} + \mathring{q}({}^A\mathbf{t}_B)$	pA = qB2A.rotate(pB) + tB2A
${}^A\mathbf{R}_B = {}^B\mathbf{R}_A^T$	rB2A = rA2B.' = transpose(rA2B)
${}^A\mathring{q}_B = {}^B\mathring{q}_A^*$	qA2B = qB2A' = qB2A.conj() = conj(qB2A) = qB2A.ctranspose() = ctranspose(qB2A)

Simbolni zapis	Znakovni zapis
${}^A\mathbf{t}_B = -{}^A\mathbf{R}_B {}^B\mathbf{t}_A$	<code>tB2A = -rB2A*tA2B</code>
$\mathring{q}({}^A\mathbf{t}_B) = {}^B\mathring{q}_A \mathring{q}(-{}^B\mathbf{t}_A) {}^B\mathring{q}_A^{-1}$	<code>tB2A = qA2B.rotate(-tA2B) = qB2A.conj().rotate(-tA2B)</code>

1 Uvod

Trenutki ujeti v slikah ne omogočajo le hranjenje spominov v foto albumih, temveč je možno, s primerno strojno obdelavo, slike uporabiti tudi kot informacijo za zaznavanje okolja oz. stanja v okolju. Velika večina visoko-razvitih živalskih vrst, med katere spada tudi človek, uporablja oči kot enega izmed osnovnih čutil za zaznavanje okolja. Čut vida nam omogoča, da se orientiramo in gibljemo med ovirami v prostoru, si zgradimo predstavo o okolju, razpoznamo in premikamo objekte ter opravljamo razna opravila, posnemamo gibe in se učimo novih spretnosti, pišemo in beremo, itd. Uspešnost uporabe vida za zaznavanje okolja in opravljanje najrazličnejših nalog pri živalih že dolgo navdihuje raziskovalce po vsem svetu, da bi razvili robota, ki bi bil zmožen opravljati opravila s strojnim vidom, na podoben način kot človek. Na določenih področjih vodenja se je uporaba strojnega vida dandanes že dobro uveljavila, predvsem v industrijskih okoljih, kjer je možno primerno prirediti delovni prostor. Čeprav so računske zmogljivosti sodobnih računalniških sistemov v zadnjih letih močno narasle, je strojna obdelava slik še vedno zelo zahtevna naloga. Na eni strani slika ponuja ogromno količino informacij, a signali, ki nas zanimajo, običajno niso neposredno merljivi. Znan je rek, da slika pove več kot tisoč besed. Z rekom se lahko malce poigramo in izračunamo število znakov, ki jih lahko zapišemo v spominskem prostoru ene sivinske digitalne slike dimenzije 512 krat 512 slikovnih pik, katerih zaloga vrednosti šteje 256 vrednosti¹. Če upoštevamo, da en znak zapišemo z osmimi biti (kar omogoča zapis 256 različnih znakov), ugotovimo, da na omejeni prostor lahko zapišemo kar

¹Čeprav je predpostavljena dimenzija slike za današnje čase relativno majhna, pa se na področju strojne obdelave slik v realnem času pogosto uporabljajo slike v tem dimenzijskem rangi, saj je strojna obdelava slik računsko zelo zahtevna.

262 144 znakov, kar je približno 39 000 besed oz. kar 75 strani besedila². Če pa dodamo še dejstvo, da črke običajno prikažemo s simboli, ki so tudi slike, lahko zapademo v neskončno filozofsko rekurzijo — prekinimo.

Ne le, da najdemo digitalno kamero skoraj v vsakem žepu³, temveč so bile z leti razvite tudi najrazličnejše metode za strojno obdelavo slik in razpoznavanje vzorcev, kar je omogočilo uporabo kamere tudi kot senzor za zaznavanje okolja in izvajanje meritev na podlagi slikovne informacije. V tem delu je predstavljena možnost uporabe kamere za izvedbo povratnozančnega vodenja oz. regulacije robotskih sistemov. Glavnina dela je osredotočena na izvedbo vodenja s strojnim vidom na avtonomnih mobilnih sistemih v zaprtih strukturiranih neprirejanih okoljih. Delo se osredotoča na reševanja določenih tipičnih problemov, ki nastopajo pri izvedbi avtonomnih mobilnih sistemov, kot je vodenje po trajektoriji. Pri izvedbi smo zasledovali predvsem idejo, da želeno nalogo podamo na sliki in poskusimo rešiti tudi problem vodenja neposredno na sliki. V delu je predstavljena izvedba regulatorja, ki omogoča reševanje problema sledenja trajektoriji neposredno na podlagi meritev na sliki. Prikazan je način načrtovanja naloge oz. poti neposredno na sliki in spremljanje vodenja preko slike. Izvedba regulacije je zasnovana z uporabo modela, ki omogoča izvedbo dvoprostostnega vodenja z direktno in povratno zanko. V delu je prikazana tudi izvedba prediktivnega vodenja, ki ga uporabimo pri izvedbi povratne zanke z upoštevanjem modela pogreška. Pri izvedbi vodenja s strojnim vidom obravnavamo različne postavitve kamere in mobilnega sistema oz. robota. Predmet zanimanja v tem delu so predvsem nekanonične postavitve. Prikazana je tudi možnost prenosa izvedenega vodenja mobilnih sistemov po trajektoriji na vodenje robotske roke po trajektoriji.

Zgradba dela je sledeča. V poglavju 2 je predstavljen pregled področja vodenja na osnovi slike. V poglavju 3 so podani nekateri matematični opisi kamere in predstavljene metode za ocenitev notranjih in zunanjih parametrov kamere. Sledi poglavje 4 o strojni obdelavi slik in uporabi lokalnih značilnih točk za izvedbo ocenjevanja lege kamere oz. mobilnega sistema v okolju. V poglavju 5 je nato predstavljena izvedba opazovalnika stanj in dvoprostostnega regulatorja za vodenje mobilnega sistema po trajektoriji. Postopek načrtovanja trajektorije je prikazan

²Pri izračunu je bilo upoštevano, da ena stran v povprečju vsebuje 3500 znakov s presledki oz. 3000 znakov brez presledkov oz. 520 besed.

³Dandanes ima večina mobilnih naprav vgrajeno vsaj eno digitalno kamero.

v poglavju 6. Predstavljen pristop vodenja mobilnega sistema je v poglavju 7 prenesen še na vodenje robotske roke po trajektoriji. V zaključku (poglavje 8) so podane sklepne misli in nakazane možnosti za nadaljnji razvoj in izboljšave.

2 Pregled področja

Vodenje na osnovi slike (angl. „*visual servoing*”) je interdisciplinarno področje, ki združuje: avtomatiko, robotiko, strojni vid, obdelavo signalov, razpoznavanje vzorcev, modeliranje, simulacijo in vodenje. Sistematičen pregled področja, opis problematike in metodologijo reševanja so podali P. Corke [1–3], F. Chaumette in S. Hutchinson [4, 5], G. Oriolo [6], E. Malis [7–9], D. Kragic [10]. V nadaljevanju tega poglavja so na kratko podani osnovni pojmi in relacije, ki se pogosto pojavljajo pri vodenju na osnovi slike. Predstavljena so tudi povezana področja, kot je strojni vid in vodenje avtonomnih mobilnih sistemov — pri čemer smo v določeni meri sledili definicijam, ki so podane v zgoraj omenjenih delih. Vodenje na osnovi slike je bilo uspešno uporabljeno v najrazličnejših aplikacijah, kot so: avtonomna vožnja in lokalizacija vozil [11–20], brezpilotno letenje in avtonomno pristajanje letal [21–23], avtonomno letenje in lebdenje kvadrokopterjev [24–27], vodenje satelitov [28], vodenje robotskih rok [1, 29–31], za pomoč kirurgom pri operacijah [32, 33], obiranje in obrezovanje sadnih dreves [34] in še in še. O področju vodenja na osnovi slike in povezanih področjih je bilo napisanih več knjig — med pomembnejšimi so [2, 3, 35–46] (slika 2.1).

Cilj vodenja na osnovi slike (VOS) je izvajanje robotizirane naloge (regulacija gibanja robota) z uporabo slikovne informacije v povratni zanki [4, 5]. Osnovna sestavna dela vsakega sistema za vodenje na osnovi slike sta torej robotski mehanizem in kamera. Vendar na področje VOS ne spada vsak sistem, ki ga sestavlja kombinacija kamere in robota. Obstajajo namreč sistemi, ki s kamero odčitajo lego objektov v prostoru, robotska manipulacija s temi objekti pa nato poteka odprtozankno, brez nadaljnjih odčitkov s pomočjo kamere — pristop je bolj znan pod imenom *poglej in premakni* (angl. „*look-and-move*”) [2, 47]. Za VOS je torej tipično, da je kamera zaprta v povratno zanko, ki stalno *opazuje* in *ocenjuje* dogajanje v delovni okolici. Za VOS lahko torej rečemo, da spada v širšo skupino



Slika 2.1: Naslovnice nekaterih pomembnejših knjig s področja strojnega vida, avtonomnih mobilnih sistemov, robotike in vodenja na osnovi slike.

vodenja s strojnim vidom, v katerega prištevamo tudi pristop *poglej in premakni*. Pravkar vpeljeni termini so natančno podani v definicijah 2.1, 2.2 in 2.3.

Definicija 2.1. *Vodenje s strojnim vidom* je vsakršen način vodenja, ki za zaznavanje okolja uporablja slikovno informacijo.

Definicija 2.2. *Vodenje na osnovi slike (VOS)* je način regulacije robotskega mehanizma, ki v povratni zanki uporablja slikovno informacijo.

Definicija 2.3. *Pristop pogledj in premakni* je način vodenja robotskega mehanizma, kjer se na podlagi slikovne informacije določi le željeno (referenčno) stanje sistema.

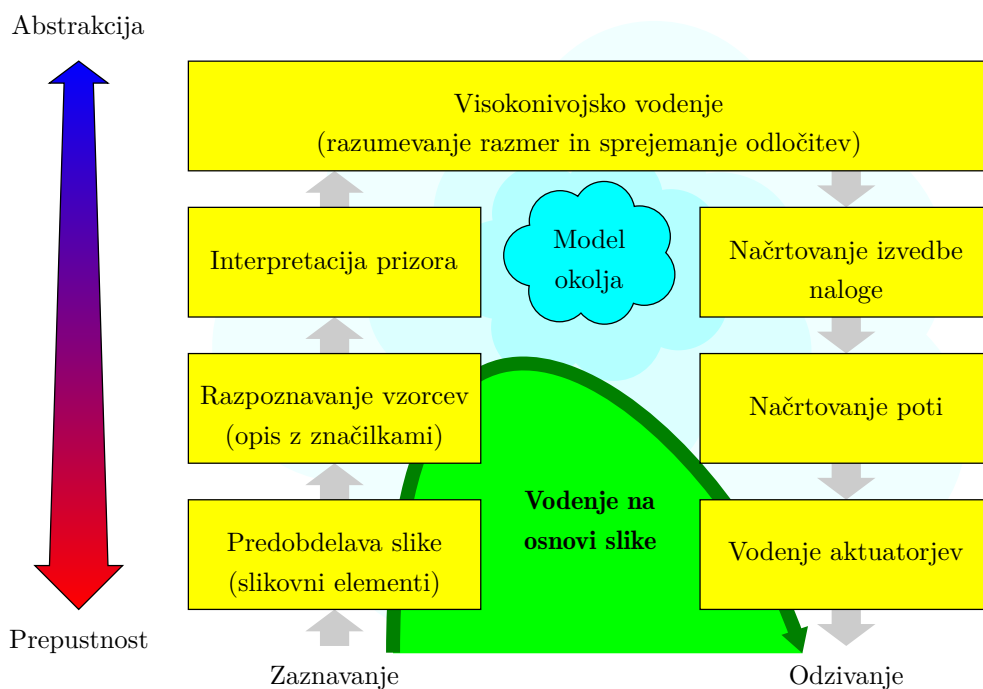
Morda velja poudariti, da sta pristopa VOS in *poglej in premakni* zasnovana za reševanje robotizirane naloge, medtem ko *vodenje s strojnim vidom* načeloma lahko zajema tudi postavitev, ki ne vključuje vodenja robotskega mehanizma. Primer takšnega vodenja je sistem za samodejno ostrenje slike pri kameri. V mislih imamo *pasivni način* samodejnega ostrenja, kjer s pomočjo informacije o ostroti slike, ki jo dobimo s primerno analizo slike (npr. z uporabo *direktne kosinusne transformacije* [48]), reguliramo ostrost slike z spreminjanjem razdalje med zaslonom in lečo.

V tak sklop vodenja sodi tudi primer pametne prometne signalizacije, kjer se uporabljajo kamere za ugotavljanje obremenjenosti cest in stanja v prometu, kar omogoča samodejno prilagajanje prometne signalizacije voznim razmeram, s čimer se lahko izboljša prometna pretočnost in hitro prilagajanje signalizacije izjemnim dogodkom (nesreče, reševalna vozila, itd.). Možna bi bila tudi izvedba regulacije ogrevanja prostora, kjer s kamero zaznavamo prisotnost ljudi v prostoru in ostala stanja okolja, kot je odprtost oz. zaprtost oken in vrat. To so le nekateri primeri, kjer s strojnim vidom ne reguliramo robotskega mehanizma. Velja še opozoriti, da k *vodenju s strojnim vidom* štejemo vse pristope, ki uporabljajo slikovno informacijo, ne glede na način, kako je bila ta informacija iz okolja zajeta — bodisi s kamero bodisi na kakšen drugačen način. Pri običajni kameri predstavlja slika svetlost zajete svetlobe na zaslonu kamere, infrardeča kamera podaja temperaturo slikanih predmetov, pri t. i. *globinski kameri* podaja slika oddaljenost predmetov v okolju od slikovne ravnine kamere, ultrazvočna slika predstavlja prečni presek slikanega objekta, itd. V vseh primerih gre za projekcijo dvo- ali več-dimenzionalnega prostora na dvodimenzionalno slikovno ravnino. Metode strojne obdelave slik lahko uporabimo na slikah, ki so nastale na različne načine, a v splošnem ne moremo vedno uporabiti istega postopka obdelave, saj so slike nastale z različnimi projekcijskimi modeli.

V tem delu so obravnavane le metode reševanja problema VOS, vendar so nekateri predstavljeni pristopi uporabni tudi splošneje. Glede na lastnosti lahko pristope VOS razdelimo v različne skupine.

Pristope VOS je možno uporabiti v raznovrstnih aplikacijah, zato smo priča mnogim različnim postavitvam in možnostim [2, 3]. Postavitve ločimo glede na število uporabljenih kamer. Lahko je celotna izvedba rešitve VOS osnovana na uporabi ene same kamere, lahko izvedba zajema uporabo dveh ali več kamer — govorimo o *mono* in *stereo* postavitvah kamere. Uporaba dveh ali več kamer omogoča določitev globine, ocenitev 3D-položaja točk in 3D-rekonstrukcijo prostora [49]. V primeru, da se objekti v okolju ne premikajo oz. se premikajo počasi, lahko s primernim gibanjem kamere dosežemo tudi 3D-rekonstrukcijo prostora z uporabo zgolj ene kamere — t. i. pristop rekonstrukcije prizora na osnovi gibanja (angl. „*structure from motion*”) [50]. Ključni problem, ki se pojavlja pri rekonstrukciji 3D-prizora, je določevanje ustreznih parov točk med slikami, ki pripadajo isti točki v prostoru. Poleg pasivnih kamer, obstajajo še aktivne,

ki delujejo tako, da projicirajo v prostor določen vzorec in odčitavajo sliko tako *osvetljenega* prizora [51]. To jim omogoča nekoliko olajšano iskanje ustreznih parov točk med slikami. V zadnjem času smo priča pojavu kamer, ki omogočajo določitev globine z merjenjem časa preleta (angl. „*time-of-flight*”) [52]. Sistem kamer je lahko vpet v sam vrh robotskega sistema — t. i. postavitev *oko na dlani* (angl. „*eye-in-hand*”) — ali pa je sistem kamer nameščen v okolju kot zunanji opazovalec robotskega sistema — t. i. postavitev *oko ob dlani* (angl. „*eye-to-hand*”). Robotski sistem je lahko bodisi robotski manipulator bodisi mobilni sistem. Nadalje lahko postavitve delimo še glede na splošne delitve, ki se pojavljajo v robotiki in avtonomnih mobilnih sistemih: glede na holonomičnost sistema, število prostostnih stopenj, obliko robotskih sklepov, obliko mehanizma, itd. Vodenje na osnovi slike lahko razdelimo tudi glede na lastnosti uporabljene kamere — glede na tip, obliko in velikost senzorja, tip zaslonke, obliko sistema leč, svetlobni spekter, itd. Vsekakor velja poudariti, da so poleg omenjenih postavitev možne različne kombinacije osnovnih postavitev in možnosti. Kakorkoli, zadana naloga običajno omejuje možne postavitve za rešitev problema.



Slika 2.2: Zgradba vodenja na osnovi slike¹.

Zaradi same narave sistema, predstavlja izvedba vodenja s strojnim vidom zelo zahteven problem. Slika predstavlja izredno bogato informacijo o okolju, a informacija potrebna za izvedbo vodenja običajno ni neposredno merljiva. Slika nastane s projekcijo prizora, zato je informacija o okolju vedno popačena, kar otežuje slikovno merjenje. Sistemi vodenja s strojnim vidom so zato običajno zgrajeni iz mnogo podsistemov, ki jih lahko v grobem razdelimo na sistem za strojno obdelavo slike in sistem za vodenje. Prvi pristopi izvedbe VOS so se reševanja problema lotili zelo visoko-nivojsko (slika 2.2). S postopki strojnega vida — predobdelava slike, razpoznavanje vzorcev, interpretacija prizora — se prizor na sliki predstavi v abstraktni obliki, nato pa se s pomočjo razumevanja razmer sprejmejo ustrezne odločitve, ki se nato preko različnih podsistemov — načrtovanje izvedbe naloge, načrtovanje poti, vodenje izvršnih členov — prevedejo v ustrezne regulirne akcije. Takšna abstraktna obravnava problema ima nizko prepustnost, zato na področju vodenja robotskih sistemov praktično ni možna izvedba v realnem času. Za izboljšanje frekvence delovanja so sistemi VOS običajno izvedeni tako, da je regulacijski zakon izpeljan neposredno na podlagi nizkonivojske obdelave slike [2]. To je na sliki 2.2 predstavljeno kot bližnjica, saj se s takšno zgradbo izognemo izvedbi zahtevnih podsistemov za abstrakcijo problema. Vse potrebno znanje o okolju običajno zberemo v ustreznem modelu sistema, ki ga lahko uporabimo tako pri načrtovanju kot izvedbi vodenja.

Oblike VOS se razlikujejo glede na definicijo pogreška. Cilj vsakega vodenja na osnovi slike je minimizacija pogreška $e(t)$. Le-tega lahko običajno podamo v splošni obliki kot razliko med želenim vektorjem značilk $s_r(t)$ in trenutnim vektorjem značilk $s(y(t), \zeta)$ [4]:

$$e(t) = s_r(t) - s(y(t), \zeta). \quad (2.1)$$

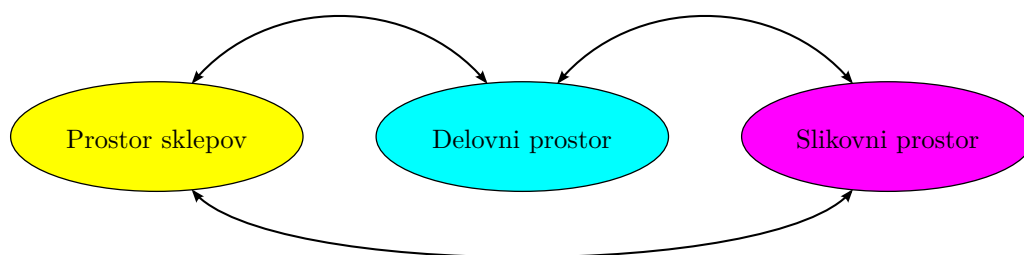
V enačbi (2.1) predstavlja vektor $y(t)$ meritve opravljene na sliki (npr. slikovne koordinate določenih vzorcev, oblika vzorcev na sliki, ...) na podlagi katerih izračunamo vektor K značilk $s(y(t), \zeta) \in \mathbb{R}^K$. Vektor ζ podaja dodatne parametre, ki opisujejo sistem (npr. notranji parametri kamere, model opazovanega objekta, dodatno znanje o sistemu in okolju, ...) — običajno so ti parametri med izvajanjem VOS nespremenljivi. Glede na enačbo (2.1) je oblika pogreška odvisna od izbire vektorja značilk.

¹Prirejeno po [2].

Preden si ogledamo osnovno razdelitev pristopov VOS, se vprašajmo: „Kaj sploh so značilke?” Lahko jih podamo z definicijo 2.4.

Definicija 2.4. *Značilke so neodvisne spremenljivke, ki opisujejo določen signal, vzorec ali stanje v okolju.*

V primeru slik lahko z značilkami opišemo barvo, točke, črte, robove, kote, ploskve, konture, obrise, teksturo, itd. Značilke lahko tako predstavljajo barvne nivoje; točke, premice, daljice, kroge, elipse in ostale stožnice; splošne krivulje, trirazsežne geometrijske objekte, geometrijske momente, odvode, itd. Nekaj primerov izpeljave značilk je prikazanih tudi v nadaljevanju. Značilke predstavljajo zgoščen zapis določene informacije in jih lahko uporabimo za različne namene. Na področju strojnega vida se značilke pogosto uporabljajo za iskanje parov točk med slikami. V primeru VOS nas zanima predvsem nabor značilk, na podlagi katerih lahko določimo regulacijski pogrešek, za kar uporabimo drugačne značilke kot za iskanje parov točk med slikami. Izbira značilk je torej povezana s problemom, ki ga želimo rešiti. Izkaže se, da je izbira značilk eden izmed ključnih problemov, ki se pojavljajo pri izvedbi VOS. Kajti šele s primerno izbiro značilk lahko dosežemo zadovoljivo delovanje sistema VOS.



Slika 2.3: Osnovni prostori.

Značilke so lahko podane v različnih prostorih (slika 2.3): npr. v trirazsežnem delovnem prostoru ali v dvorazsežnem slikovnem prostoru. Glede na obliko vektorja značilk, ki vpliva na obliko pogreška (2.1), ločimo naslednje oblike VOS [2]: *položajno vodenje na osnovi slike* (angl. „*position based visual servoing*”), *direktno vodenje na osnovi slike* (angl. „*image based visual servoing*”), *lokalno*

vodenje na osnovi slike in hibridno vodenje na osnovi slike (angl. „*hybrid visual servoing*”); ki jih podajajo definicije 2.5, 2.6, 2.7 in 2.8.

Definicija 2.5. *Položajno vodenje na osnovi slike (PVOS)* je oblika VOS, kjer je pogrešek določen na podlagi veličin, ki so podane v trirazsežnem delovnem prostoru.

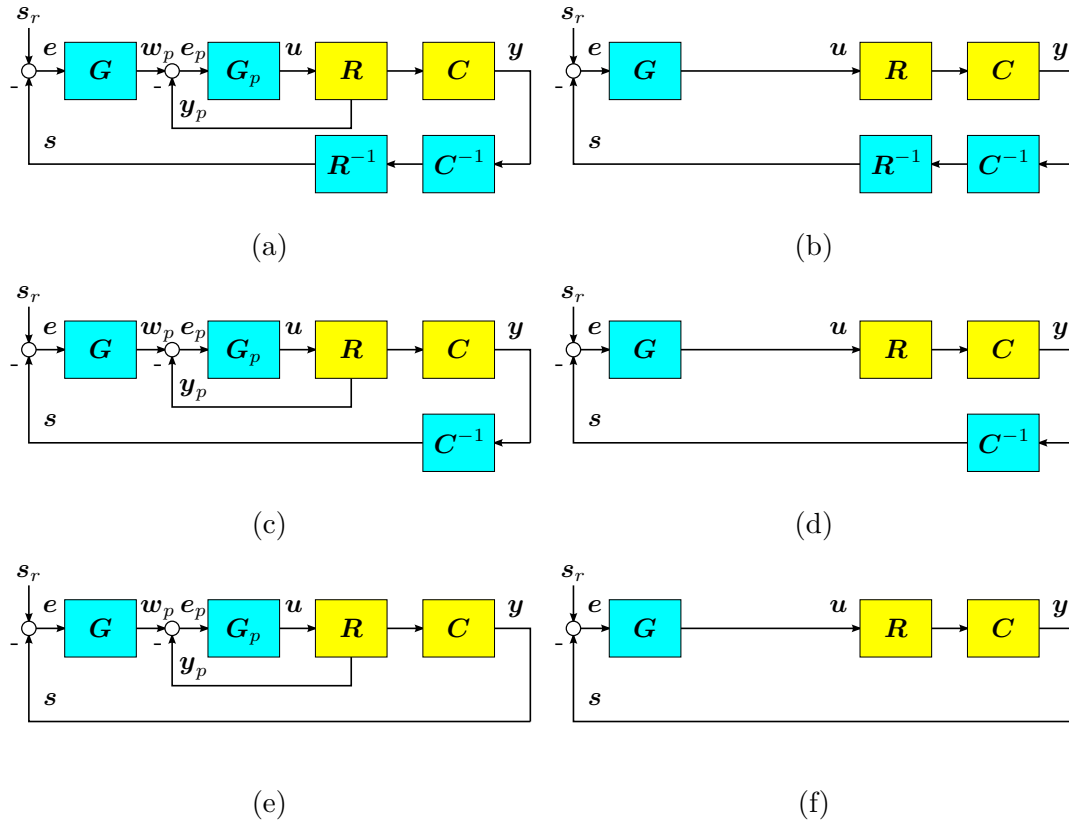
Definicija 2.6. *Direktno vodenje na osnovi slike (DVOS)* je oblika VOS, kjer je pogrešek določen na podlagi veličin, ki so podane neposredno na dvorazsežni sliki.

Definicija 2.7. *Lokalno vodenje na osnovi slike (LVOS)* je oblika VOS, kjer je pogrešek določen na podlagi veličin, ki so podane v notranjih koordinatah sklepov.

Definicija 2.8. *Hibridno vodenje na osnovi slike (HVOS)* je oblika VOS, kjer je pogrešek določen na podlagi veličin, ki so podane tako v trirazsežnem delovnem prostoru kot neposredno na dvorazsežni sliki. Gre za kombinacijo PVOS in DVOS.

Z značilkami je definiran regulacijski pogrešek. Pri primerni izbiri značilk, pogrešek izgine, ko trenutne značilke sovpadajo z želenimi značilkami. Predvsem zaradi perspektivne projekcije optimalni pogrešek za opravilo določene naloge običajno ni neposredno merljiv, temveč ga izrazimo posredno s primerno izbiro značilk. V določenih primerih lahko postane predstavitev pogreška z značilkami dvoumna ali slabo pogojena, kar lahko privede do stanja, kjer izničenje pogreška značilk še ne pomeni izničenja pogreška v delovnem prostoru. Pojavi se lahko problem neoptimalnih ali celo neželenih prehodnih pojavov, ki lahko privedejo do destabilizacije sistema. Kot bomo videli v nadaljevanju, je problem še posebej pereč pri izvedbi DVOS.

Pri PVOS (definicija 2.5, sliki 2.4c in 2.4d) predstavlja vektor značilk $\mathbf{s}$ meritve okolja v trirazsežnem prostoru, ki je hkrati tudi delovni prostor sistema [53, 54]. Ker so veličine v trirazsežnem prostoru določene na podlagi projekcije tega prostora na dvorazsežno sliko, je to VOS. Z drugimi besedami, na podlagi meritev na sliki pri PVOS s pomočjo modela ocenimo veličine v 3D-prostoru. Ker je pri PVOS pogrešek podan v delovnem prostoru, lahko običajno z regulacijo enostavno dosežemo optimalne premike. Slaba lastnost PVOS je predvsem v dejstvu, da postopek zahteva točno kalibracijo kamere, sicer lahko pride do velikih



Slika 2.4: Konceptualni prikaz različnih oblik VOS. Na slikah (a) in (b) je LVOS, na slikah (c) in (d) je PVOS, na slikah (e) in (f) je DVOS. Levi stolpec prikazuje kaskadne izvedbe VOS s pomožno regulacijsko zanko, medtem ko desni stolpec prikazuje neposredne izvedbe. $\mathbf{R}$ je model robotskega sistema, $\mathbf{C}$ je model kamere, $\mathbf{G}$ in $\mathbf{G}_p$ je sta glavni in pomožni regulator, $\mathbf{R}^{-1}$ in $\mathbf{C}^{-1}$ sta inverzna modela robotskega sistema in kamere; $\mathbf{s}_r$ je vektor želenih vrednosti značilnk, $\mathbf{s}$ je vektor trenutnih vrednosti značilnk, $\mathbf{y}$ je vektor meritev na sliki, $\mathbf{u}$ je regulirni signal, $\mathbf{e}$ je regulacijski pogrešek; $\mathbf{w}_p$, $\mathbf{y}_p$ in $\mathbf{e}_p$ so signali pomožnega regulatorja.

pogreškov, ki jih regulator ne more odpraviti. Velik problem pri PVOS predstavlja ocenitev globine prizora na podlagi slike (projekcije) okolja. Za ocenjevanje globine se tako pogosto uporablja stereo postavitev oz. se globino oceni s primernim opazovalnikom stanj (npr. s Kalmanovim filtrom) [53, 55].

Medtem so pri DVOS (definicija 2.6, slike 2.4e in 2.4f) elementi vektorja značilk $\mathbf{s}$ podani z neposrednimi meritvami na dvorazsežni sliki [46, 56–59]. Torej je glede na enačbo (2.1) pogrešek pri DVOS podan kot razlika med trenutnimi in želenimi vrednostmi v dvorazsežnem prostoru (neposredno na sliki), medtem ko je pri PVOS podan kot razlika med na podlagi slike ocenjenimi vrednostmi v trirazsežnem prostoru. Pri tem velja poudariti, da se pri obeh pristopih uporablja dvorazsežni slikovni signal na podlagi katerega vodimo sistem v trirazsežnem delovnem prostoru. Izvedbe DVOS so običajno manj občutljive na točnost kalibracije sistema, a ker pogrešek ni definiran v trirazsežnem delovnem prostoru, je običajno dosegljiva le lokalna konvergenca. Odzivi pri regulaciji sistema z DVOS običajno niso optimalni, pride pa lahko tudi do precej nezaželenih ali celo nepredvidljivih odzivov. Pojavlja se problem t. i. *samovoljnosti kamere* (angl. „*camera retreat*”) [4], kjer sistem, navkljub enostavnemu želenemu premiku, naredi zelo zapleten premik, ki pogosto ni v skladu s pričakovanji. Zelo znan je primer, ko želimo, da se kamera le zavrti za določen zasuk okoli optične osi, vendar se določene izvedbe DVOS v takem primeru odzovejo z neželenim oddaljevanjem in približevanjem v smeri optične osi; pri določenih zasukih pa postane sistem celo nestabilen.

V tem delu smo vpeljali še LVOS (definicija 2.7, slike 2.4a in 2.4b), ki ga na področju VOS le redko srečamo in se ga običajno niti ne navaja — tu ga omenjamo le zaradi celovitejšega opisa. Pri pristopu gre za to, da na podlagi slike določimo pogrešek v notranjih koordinatah (sklepih) robotskega manipulatorja. Še bolj kot pri PVOS je postopek občutljiv na kakovost kalibracije sistema in zato s takšnim načinom vodenja običajno težko dosežemo izničenje pogreška v delovnem prostoru. V primerjavi z ostalima pristopoma je izvedba regulacijskega zakona v tem primeru preprostejša, saj lahko neposredno reguliramo posamezne sklepe.

Vsekakor velja poudariti, da so ne glede na obliko VOS, meritve $\mathbf{y}(t)$ vedno opravljene na dvorazsežni sliki trirazsežnega delovnega prostora. Poleg omenjenih načinov VOS obstaja še možnost, ki združuje dobre lastnosti vseh omenjenih oblik VOS in jo zato imenujemo *hibridno vodenje na osnovi slike* (definicija 2.8) — običajno združujemo le pristope PVOS in DVOS. Pri HVOS so veličine v vektorju

značilke $\mathbf{s}$ delno izbrane v trirazsežnem delovnem prostoru in delno v dvorazsežni slikovni ravnini. Zaradi takšne hibridne zasnove vektorja značilke pravimo, da so veličine izbrane v 2,5–razsežnem (2,5 D) prostoru [30, 60]. To ni edina možna oblika HVOS. Med pomembnejše spadata predvsem *preklopno hibridno vodenje na osnovi slike* (angl. „switched visual servoing”) [61] in *razklopljeno vodenje na osnovi slike* (angl. „partitioned visual servoing”) [5], ki ju podajata definiciji 2.9 in 2.10.

Definicija 2.9. *Preklopno (hibridno) vodenje na osnovi slike (PHVOS) je oblika HVOS, ki vsebuje izbirni mehanizem, ki glede na stanje sistema preklaplja med PVOS in DVOS obliko regulacije.*

Definicija 2.10. *Razklopljeno vodenje na osnovi slike (RVOS) je oblika VOS, ki definira vektor pogreška tako, da je v idealnem primeru vsaka regulirna veličina povezana edino z eno veličino pogreška.*

Preklopno VOS (definicija 2.9) je sestavljeno iz dveh regulatorjev, kjer je eden zasnovan v obliki PVOS, drugi pa v obliki DVOS. Med obratovanjem je vedno aktiven le en izmed regulatorjev. Regulacijski sistem sestavlja še preklopni mehanizem, ki sproti preverja, katera regulacijska struktura je v dani situaciji bolj optimalna in skrbi za mehke preklope med regulatorjema. Na ta način lahko uspešno rešimo problem VOS, saj lahko z DVOS običajno izničimo pogrešek, ko se nahajamo v primerno majhni okolici referenčne lege, medtem ko s PVOS dosežemo dokaj optimalne premike, četudi se nahajamo daleč stran od referenčne lege. Torej se lahko izognemo problemu lokalnih minimumov, ki se pojavlja pri DVOS, in problemu izničenja pogreška v ustaljenem stanju, ki se pojavlja pri PVOS. Takšna izvedba VOS se je izkazala za uspešno, vendar je sama struktura algoritma precej zapletena, saj je sestavljena iz treh sklopov.

Na področju VOS je bilo veliko raziskovalnega truda vloženega v razvoj razklopljenega VOS (definicija 2.10). Pri tej obliki v idealnem primeru vsaka značilka vpliva na natanko eno regulirno veličino. Ker v večini primerov ni enostavno ali celo možno najti takšnega vektorja značilke, se pogosto temu idealu želimo le približati. Tako ima vsaka značilka največji vpliv le na eno samo regulirno veličino. S tega vidika je še posebno izrazito dejstvo, da predstavlja izbira vektorja značilke $\mathbf{s}$ enega večjih izzivov na področju VOS.

Oglejmo si sedaj, kako je izvedena osnovna regulacija pri VOS². Predstavljene izvedbe so dokaj splošne in načeloma veljajo za vse ravnokar omenjene oblike VOS. Zaenkrat se bomo omejili le na primer, ko je vektor referenčnih značilnk v enačbi (2.1) časovno nespremenljiv ($\mathbf{s}_r(t) = \mathbf{s}_r = \text{konst.} \in \mathbb{R}^K$, kjer je K dolžina vektorja značilnk). Eden izmed najpogostejše uporabljenih regulacijskih pristopov na področju VOS je t. i. *regulacija po hitrosti*. Regulacijski zakon temelji na poznavanju povezave med hitrostjo sprememb značilnk in hitrostjo sprememb regulirnih akcij. Lahko si predstavljamo, da imamo opravka s postavitvijo kamere na vrh robotskega sistema, čeprav je predstavljen opis uporaben tudi bolj splošno. Predpostavimo, da regulirni vektor predstavljajo tri translatorne hitrosti $\mathbf{v}^T(t) = [v_x(t), v_y(t), v_z(t)]$ in tri kotne hitrosti $\boldsymbol{\omega}^T(t) = [\omega_x(t), \omega_y(t), \omega_z(t)]$, ki jih predstavimo v razširjenem vektorju $\mathbf{u}^T(t) = [\mathbf{v}^T(t), \boldsymbol{\omega}^T(t)] \in \mathbb{R}^6$. Hitrost premika $\mathbf{u}^T(t)$ je povezana s hitrostjo spreminjanja značilnk $\dot{\mathbf{s}}^T(t)$ [4]:

$$\dot{\mathbf{s}}(t) = \mathbf{L}(t)\mathbf{u}(t), \quad (2.2)$$

kjer je $\mathbf{L}(t) \in \mathbb{R}^K \times \mathbb{R}^6$ Jacobijeva matrika, ki jo na področju VOS bolj znana kot *interakcijska matrika*. Z združitvijo enačb (2.1) in (2.2) neposredno sledi izraz za hitrost spreminjanja pogreška [4]:

$$\dot{\mathbf{e}}(t) = -\mathbf{L}(t)\mathbf{u}(t). \quad (2.3)$$

Cilj načrtovanja regulatorja je poiskati regulirni signal $\mathbf{u}(t)$, ki minimizira vrednost pogreška $\mathbf{e}$. Ob zahtevi za eksponentialno padanje pogreška ($\dot{\mathbf{e}}(t) = -g\mathbf{e}(t)$, $g > 0$), pridemo do regulacijskega zakona:

$$\mathbf{u}(t) = g\mathbf{L}^\dagger(t)\mathbf{e}(t). \quad (2.4)$$

kjer je $\mathbf{L}^\dagger(t)$ Moore-Penrosov psevdoinverz interakcijske matrike $\mathbf{L}(t)$. Če je matrika $\mathbf{L}(t)$ polnega ranga, je njena psevdoinverzna matrika podana z izrazom $\mathbf{L}^\dagger(t) = (\mathbf{L}^T(t)\mathbf{L}(t))^{-1}\mathbf{L}^T(t)$. Če je interakcijska matrika kvadratna (število značilnk $K = 6$) in ni singularna ($\det(\mathbf{L}(t)) \neq 0$), se v regulacijskem zakonu (2.4) operacija izračuna psevdoinverzne matrike poenostavi v izračun običajne inverzne matrike $\mathbf{L}^{-1}(t)$.

²Predstavljene definicije so povzete po delih [4, 5].

V praksi vrednost interakcijske matrike oz. njena inverzna vrednost ni točno znana. Uporabimo lahko le oceno interakcijske matrike $\hat{\mathbf{L}}(t)$ ali oceno inverzne interakcijske matrike $\hat{\mathbf{L}}^\dagger(t)$. Tako je regulacijski zakon (2.4) dejansko:

$$\mathbf{u}(t) = g\hat{\mathbf{L}}^\dagger(t)\mathbf{e}(t). \quad (2.5)$$

Oceno $\hat{\mathbf{L}}^\dagger(t)$ lahko določimo na različne načine. Lahko jo določimo na podlagi ocene trenutnega stanja sistema $\hat{\mathbf{L}}^\dagger(t) = \hat{\mathbf{L}}_t^\dagger(t)$ ali pa jo ocenimo na podlagi referenčnega stanja sistema $\hat{\mathbf{L}}^\dagger(t) = \hat{\mathbf{L}}_r^\dagger(t)$. Popularna in dokaj učinkovita izbira je tudi uporaba kombinacije omenjenih možnosti na način: $\hat{\mathbf{L}}^\dagger(t) = \frac{1}{2}(\hat{\mathbf{L}}_r(t) + \hat{\mathbf{L}}_t(t))^\dagger$ [5].

Po vstavitvi regulatorja (2.5) v sistem (2.3), pridemo do diferencialne enačbe zaprtozančnega sistema:

$$\dot{\mathbf{e}}(t) = -g\mathbf{L}(t)\hat{\mathbf{L}}^\dagger(t)\mathbf{e}(t). \quad (2.6)$$

Stabilnost sistema (2.6) lahko preverimo s pomočjo Lyapunove analize. Predpostavimo Lyapunovo funkcijo $V(t) = \frac{1}{2}\mathbf{e}^T(t)\mathbf{e}(t)$ in poiščemo njen prvi odvod, pri čemer upoštevamo sistem (2.6):

$$\dot{V}(t) = -g\mathbf{e}^T(t)\mathbf{L}(t)\hat{\mathbf{L}}^\dagger(t)\mathbf{e}(t). \quad (2.7)$$

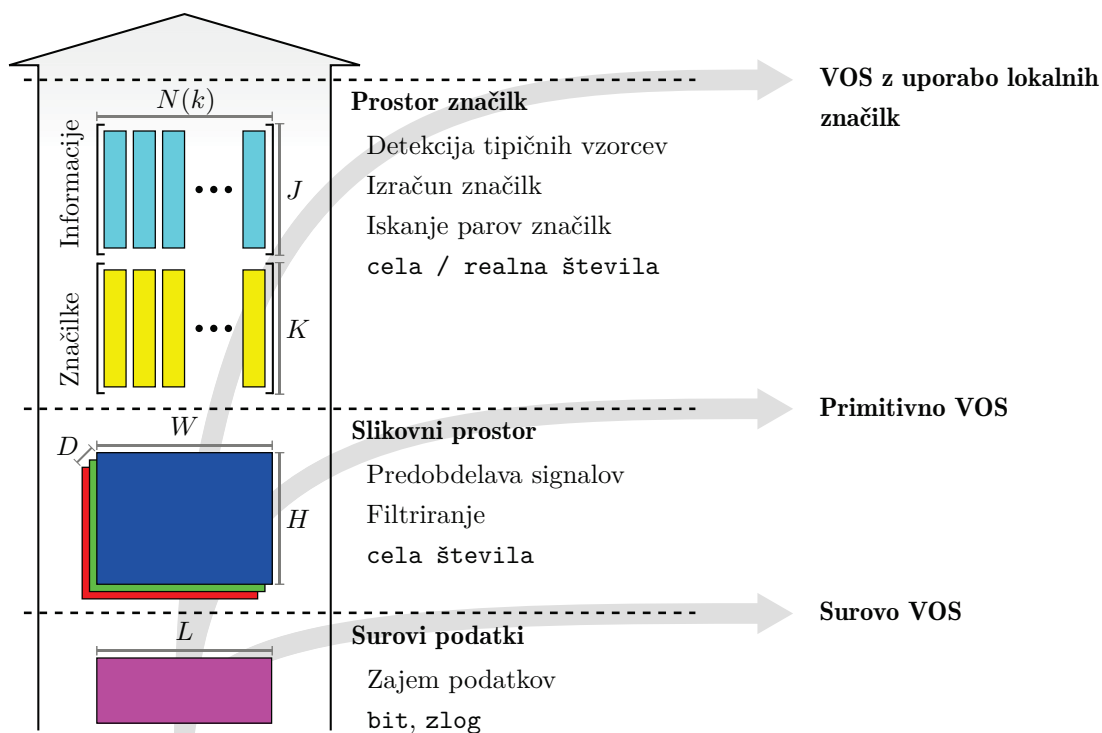
Zadosten pogoj za globalno stabilnost sistema (2.6) je negativnost odvoda Lyapunove funkcije (2.7). Negativna definitnost funkcije (2.7) je izpolnjena, ko za poljuben $\mathbf{x} \in \mathbb{R}^K$ velja:

$$\mathbf{x}^T\mathbf{L}(t)\hat{\mathbf{L}}^\dagger(t)\mathbf{x} > 0, \quad (2.8)$$

kar z drugimi besedami pomeni, da mora za stabilnost sistema biti matrika $\mathbf{L}(t)\hat{\mathbf{L}}^\dagger(t)$ pozitivno definitna. V kolikor je število značilk K enako številu regulirnih veličin (v našem primeru $K = 6$) in sta matriki $\mathbf{L}(t)$ in $\hat{\mathbf{L}}^\dagger(t)$ polnega ranga, je pogoj (2.8) izpolnjen, če le ni ocena $\hat{\mathbf{L}}^\dagger(t)$ pregroba [4].

Pri VOS pogosto srečamo kaskadno regulacijsko obliko [3, 16, 62, 63]. Pogosto notranja regulacijska zanka v kaskadni vezavi skrbi za regulacijo hitrosti motorjev, medtem ko je v zunanjo regulacijsko zanko vključen regulator za vodenje na osnovi slike, ki skrbi za doseganje želenega stanja (položaja ali hitrosti) v zunanjem koordinatnem sistemu. VOS je možno izvesti tudi z eno samo regulacijsko zanko,

kjer na podlagi izmerjenega pogreška na osnovi slike neposredno izračunamo regulirne vrednosti motorjev (izvedbe v desnem stolpcu na sliki 2.4). Takšna izvedba v mnogih primerih ni enostavno izvedljiva, saj robotski sistemi lahko ne omogočajo neposrednega krmiljenja motorjev, temveč lahko podamo le referenčne hitrosti vrtenja/premikov in/ali želene zasuke/premike. Ker ima kamera omejeno frekvenco vzorčenja in ker je obdelava slike časovno zahtevna naloga, je izvedba takšnega sistema zelo zahtevna naloga. Uporaba pomožne regulacijske zanke z relativno visoko frekvenco vzorčenja v kaskadni vezavi omogoča, da lahko sistem obravnavamo z idealizirano dinamiko (zaprtizančno dinamiko pomožne regulacijske zanke). Zunanjo regulacijsko zanko, ki jo osnujemo na principu vodenja na osnovi slike, lahko nato načrtujemo za idealizirani sistem, kar je običajno možno pri relativno nizkih frekvencah vzorčenja.



Slika 2.5: Razdelitev VOS glede na način obdelave slike.

Omenjeno je že bilo, da je VOS običajno načrtovano na podlagi preproste obdelave slike. Pri izvedbi VOS zasledimo različne nivoje obdelave slike. Med bolj razširjene pristope VOS sodita *vodenje na osnovi slikovnih značilk* (definicija 2.13) in *vodenje na osnovi primitivne slike* (definicija 2.12). V zadnjem času se

uveljavlja tudi t. i. *vodenje na osnovi surove slike* (definicija 2.11) [64], čeprav so izvedbe še vedno redke. Razmejitve med posameznimi pristopi so tako na nivoju obdelave slike, kar je prikazano tudi na sliki 2.5.

Definicija 2.11. *Vodenje na osnovi surove slike* je način VOS, kjer regulacijski pogrešek določimo neposredno iz surovih slikovnih podatkov.

Definicija 2.12. *Vodenje na osnovi primitivne slike* je način VOS, kjer regulacijski pogrešek določimo na osnovi obdelave slike na nivoju slikovnih pik, pri čemer upoštevamo lokalno okolico slikovnih elementov.

Definicija 2.13. *Vodenje na osnovi slikovnih značilnk* je način VOS, kjer regulacijski pogrešek določimo na podlagi opisa slike z značilnkami.

Pristop k *vodenju na osnovi slikovnih značilnk* je eden izmed najpogostejše uporabljenih pristopov VOS. Pri izvedbi povratno-zančnega sistema vodenja moramo meriti pogrešek med želenim in trenutnim stanjem. VOS ni v tem pogledu nobena izjema, le da predstavlja slika izjemno bogato informacijo o okolju, za namene vodenja pa nas običajno zanima le delček informacije, ki je skrit v sliki. S slike moramo tako izvleči točno določeno informacijo, na podlagi katere neposredno ali posredno določimo regulacijski pogrešek. Informacija, ki jo iščemo, se na sliki pojavi kot vzorec, vzorce pa lahko opišemo z njim lastnimi značilnostmi. Na primer, pri opazovanju položaja žoge v prostoru s kamero, na podlagi določenih značilnosti žoge, le to ločimo od ozadja; na podlagi velikosti in položaja vzorca žoge na sliki (vrednosti določenih značilnosti vzorca) lahko bolj ali manj natančno ocenimo položaj žoge v prostoru glede na kamero. Zatorej je razumljivo, da so pristopi VOS, ki so osnovani na uporabi slikovnih značilnk med najbolj razširjenimi. Določene značilnosti vzorca omogočajo, da določimo stanje objektov v prostoru, tako da vzorce opišemo z lastnostmi, na podlagi katerih lahko opravimo določeno meritev na sliki.

V določenih aplikacijah VOS je dovolj, da sledimo skozi niz slik že enemu samemu vzorcu, ki običajno predstavlja en objekt v prostoru. Zaradi omejenega vidnega kota kamere, omejenega zornega kota (oz. zornih kotov v primeru uporabe več kamer) in neizogibnih prekrivanj (okluzij), ni moč v vsakem trenutku videti celotne slike objekta. Zato določene metode razpoznavanja objekta ne delujejo, ko le-ta ni v celoti viden ali je viden pod drugim zornim kotom.

V tem delu je VOS razdeljeno še glede na način podajanja naloge. Ločimo med tremi načini podajanja naloge, ki so predstavljeni v definicijah 2.14, 2.15 in 2.16.

Definicija 2.14. Vodenje v delovnem prostoru je način podajanja naloge, kjer želeno stanje sistema podamo v delovnem prostoru sistema.

Definicija 2.15. Vodenje s slikovnim prikazom je način podajanja naloge, kjer želeno stanje sistema podamo tako, da podamo sliko pogleda v želenem stanju.

Definicija 2.16. Vodenje preko slike je način podajanja naloge, kjer predstavimo nalogo v grafični obliki na sliki.

Kamera je neinvaziven (pasiven) senzor, ki preko leče zbira svetlobo na zaslonu. Moderne digitalne kamere pretvorijo svetlobni signal (sliko), ki nastane na zaslonu, v električni signal in nato v digitalni signal (matematično je to matrika števil), ki je primeren za obdelavo na digitalnem računalniku. Pri VOS običajno želimo kratkotrajno odprtost zaslonke, tako da hitro premikajoči se objekti niso zamazani na sliki, in veliko globinsko ostrino (dosežemo jo z majhno odprtostjo zaslonke), da med vodenjem ne pride do zamegljene slike. Da lahko zadostimo omenjenima zahtevama, mora biti opazovani prizor primerno osvetljen. Poleg kamer, ki zajemajo svetlobo v vidnem svetlobnem spektru, lahko pri vodenju na osnovi slike uporabimo tudi kamere, ki zaznavajo *nevidni* del svetlobnega spektra — metodologija VOS je enaka. Pri vodenju na osnovi slike se uporabljajo različni tipi kamer: običajne perspektivne kamere in kamere s polnim vidnim kotom (angl. „*omni-directional camera*”) [20]. V tem delu smo se omejili le na uporabo običajnih perspektivnih (ozkokotnih) kamer, ki preslikajo tridimenzionalen prizor na dvodimenzionalno slikovno ravnino. V kolikor ne pride do prevelikih popačenj slike zaradi distorzije leče, lahko opišemo preslikavo tridimezionalnih točk na slikovno ravnino s perspektivnim modelom kamere (angl. „*pin-hole camera model*”) [39], sicer moramo uporabiti bolj natančen model, ki upošteva popačenja [65, 66]. Notranje in zunanje parametre izbranega modela kamere določimo s postopkom kalibracije [66–71]. Primerno modeliranje in kalibracija kamere je ključno za zadovoljivo delovanje VOS.

Pri VOS je pomembno sledenje objektom skozi sekvenco slik (film). Prvi pristopi pri sledenju objektov so se zanašali na uporabo enostavnih značk v

okolju, ki jih lahko s strojnimi vidom hitro in enolično razpoznamo, npr. uporaba barvnih značk v robotskem nogometu [72]. Ker obstajajo optimizirani algoritmi za razpoznavo značk, se le-te še vedno uporabljajo v aplikacijah kjer je le-to mogoče [24, 25]. Razviti pa so bili tudi mnogi različni napredni algoritmi, ki omogočajo sledenje objektom brez uporabe značk [73–78]. Pri sledenju objektom se pojavljajo problemi izgube in zamenjave objekta, problemi okluzij itd.

Veliko postopkov pri strojni obdelavi slik temelji na opisu slik z lokalnimi značilkami. Uporaba parov lokalnih značilnih točk omogoča določevanje globine in 3D-rekonstrukcijo prizora ter ocenjevanje lege kamere v prostoru [49, 79, 80], sledenje objektom skozi sekvenco slik [75, 81] idr. Ključni problem, ki se pojavlja pri uporabi značilk, je iskanje parov med značilkami, zato morajo značilke opisati določen vzorec na tak način, da lahko najdemo njen par na drugi sliki, če se le tak vzorec ponovi na drugi sliki in ne najdemo para, če takšnega vzorca na drugi sliki ni. Z značilkami lahko opišemo različne geometrijske strukture, ki se pojavljajo v sliki: točke, robove, kote, področja, itd. Idealno bi želeli, da so značilke invariantne na premike, skaliranje, rotacijo, perspektivno transformacijo, osvetljenost, šum in zamegljenost slike. Pomemben prelom v razvoju predstavlja algoritem SIFT [82] za samodejno iskanje primernih značilk v sivinskih slikah, opis značilk in iskanje parov med značilkami, ki so invariantne na premik, skaliranje, rotacijo in osvetljenost. Kasneje so bile predstavljene še izboljšave in pohitritve algoritma [83], kot tudi mnoge druge metode opisa slik z značilkami [84]: MSER [85], FAST [75], SUSAN [86], BRIEF [87], ORB [88], FREAK [89] idr.

Za pravilno ujemanje parov se poslužujemo algoritmov za robustno ocenjevanje parametrov modela tudi, ko podatki vsebujejo veliko napačno razvrščenih parov [90–93]. Večina teh algoritmov je izpeljana oz. temelji na algoritmu RANSAC [94]. Pri tem vpeljemo primerne modele, ki povezujejo množice značilk in omejijo množico možnih parov. Pri perspektivni transformaciji so pari med slikama omejeni s *fundamentalno matriko*, v primeru opazovanja ravnine pa s *homografijo*. Fundamentalna matrika in homografija sta pomembna geometrijska modela, na podlagi katerih je bilo zgrajenih mnogo algoritmov VOS [9, 78, 95–97]. Najnovejši trend pri VOS pa predstavljajo neposredni pristopi, ki ne zahtevajo izračuna lokalnih značilk. Ti neposredni pristopi izračunajo ustrezne regulirne akcije neposredno iz slike [64, 98].

Problem *vođenja po poti* je poleg *regulacije lege* eden izmed temeljnih problemov

v mobilni robotiki. Ločimo več različnih pristopov. Pri obravnavi vodenja po poti se omejimo le na *neholonomične* sisteme, ker večina mobilnih sistemov spada v ta razred sistemov. Obravnava neholonomičnih sistemov je zahtevna, saj imajo le-ti sistemi omejitve, ki jim preprečujejo gibanje v določenih smereh. Posledično je zahtevno tudi načrtovanje regulatorjev za vodenje takšnih sistemov [99–105]. Pokazano je bilo, da za mobilni sistem z diferencialnim dvokolesnim pogonom ne moremo izdelati časovno-nespremenljivega linearnega regulatorja za vodenje po poti — Brockettov izrek [106]. Predpostavka, ki se pogosto pojavlja pri problemu vodenja mobilnega sistema z diferencialnim dvokolesnim pogonom po poti je, da mobilni sistem med sledenjem ne miruje, kar se običajno odraža v zahtevi, da je tangencialna hitrost mobilnega sistema po absolutni vrednosti vedno večja od nič [6, 12, 16, 107]. Še pogostejše zahtevamo, da ima tangencialna hitrost vedno pozitivno vrednost, kar pomeni, da je dovoljeno le gibanje naprej — zahteva se uporablja predvsem pri mobilnih sistemih, ki imajo senzorje za zaznavanje okolja nameščene le na sprednji strani mobilnega sistema, kjer je pri vzvratni vožnji možnost zaznavanja ovir zato onemogočena.

S problemom vodenja po poti je povezan problem načrtovanja poti. Glede na načrtano pot je odvisen postopek reševanja vodenja po poti oz. obratno, način vodenja po poti določa kako moramo izvesti načrtovanje poti. Pot lahko podamo z omejenim naborom osnovnih premikov (abecedo) — pot je nato sestavljena iz črk abecede [107]. Za načrtovanje poti se pogosto uporabljajo tudi parametrične krivulje, ki so vsaj nekajkrat zvezno odvedljive [15, 108, 109]. Referenčna pot mora biti izdelana tako, da je pot optimalna za zadano nalogo. Na poti se lahko pojavijo (nepredvidene) ovire, zato se mora sistem znati oviram izogniti in morda celo spremeniti pot med samim sledenjem [108, 110–112].

Problem vodenja po poti običajno razdelimo na problem *doseganja poti* (angl. „*path-reaching*”) in problem *sledenja poti* (angl. „*path-following*”). Pri problemu sledenja poti običajno predpostavljamo, da je pogrešek med trenutno lego in želeno lego majhen. Ob upoštevanju majhnih odmikov od referenčne poti, lahko problem sledenja poti poenostavimo, npr. z metodo linearizacije modela pogreška v okolici referenčni poti [108]. Rešitev problema sledenja poti zatorej običajno ne daje dobrih rezultatov, če je začetni pogrešek velik — torej, če se nahajamo daleč od referenčne poti moramo rešiti problem doseganja poti [12]. Ko postane pogrešek med trenutno in referenčno lego dovolj majhen, lahko preklopimo iz algoritma za

doseganje poti na algoritem za sledenje poti.

Problem sledenja poti lahko zastavimo na več načinov. Med pogostejše spadata *sledenje poti* s konstantno hitrostjo [12] in *sledenje trajektoriji* (angl. „*trajectory-tracking*“) [108]. V primerjavi s sledenjem poti, pri sledenju trajektoriji želimo ne le iti po referenčni poti, temveč biti na točno določenem delu poti od točno predpisanem času.

Za problem vodenja po poti so bile uporabljene različne regulacijske metode: regulator stanj [6], prediktivni regulator [16], nelinearni regulator [105] itd. Za problem vodenja po poti je izredno primeren prediktivni regulator, saj je zadana pot običajno podana vnaprej in lahko z upoštevanjem napovedanega odziva dosežemo optimalno gibanje. Prediktivni regulator uporablja model sistema za ocenjevanje obnašanja sistema, kar omogoča določitev optimalnih regulirnih akcij, ki upoštevajo dinamiko in omejitve sistema. Prediktivno vodenje sodi torej v širše področje vodenja na osnovi modela [113–116]. Na področju prediktivnega vodenja so bile v preteklosti zasnovani različni regulacijski algoritmi: angl. „*model algorithmic control (MAC)*“ [117], prediktivni regulator na osnovi modela (angl. „*model-based predictive control (MPC)*“) [118], posplošen prediktivni regulator (angl. „*generalized predictive control (GPC)*“) [119], prediktivni funkcijski regulator (angl. „*predictive functional control (PFC)*“) [120–122] idr. Prediktivni regulator v našem primeru izdelamo tako, da lahko referenčno pot sestavimo z zlepkom parametričnih krivulj. Z Bernstein-Bézierjevimi parametričnimi krivuljami lahko napravimo približek poljubne poti [108].

Pri vodenju mobilnega sistema po poti, ko se kamera nahaja na mobilnem sistemu, se s premikanjem sistema spremeni celotna slika okolja. Glede na mobilni sistem pa se premakne tudi referenčna pot, kar vnaša določene težave. Pri vodenju na osnovi slike se pojavi še dodatna težava, saj je slika poti podvržena perspektivni transformaciji. Določene probleme lahko razrešimo tako, da referenčno pot narišemo (v kolikor je to mogoče) na podlago po kateri želimo, da se giba robot oz. iz slike razpoznamo pot [18]. Za vodenje po poti v tem primeru običajno ne potrebujemo zemljevida okolja.

Kamera ima omejen vidni kot, zato običajno ne moremo videti celotnega delovnega območja. Pri navigaciji mobilnega sistema po prostoru je koristno, če le-ta ve, kje se nahaja. V določenih primerih brez poznavanja lastne lokacije v globalnem zemljevidu ne moremo izvesti vodenja po poti. Svojo lego v prostoru

lahko avtonomni sistem ugotovi na podlagi notranjih senzorjev (npr. na podlagi enkoderjev na kolesih s pristopom odometrije), vendar je tak način uporaben le na kratke razdalje. Da se mobilni sistem lahko lokalizira potrebuje zunanjo referenco oz. zemljevid okolja — problem lokalizacije [20, 123]. Poleg lokalizacije je zemljevid okolja koristen tudi pri načrtovanju poti. Če zemljevid okolja ni na voljo, ga je potrebno zgraditi — problem gradnje zemljevida [17, 124]. Pri avtonomnih sistemih pa običajno na začetku nimamo ne zemljevida, niti ne vemo, kje smo, vendar lahko z raziskovanjem okolja raziščemo svet okoli sebe in zgradimo zemljevid ter se v tem zemljevidu lokaliziramo — problem sočasne lokalizacije in gradnje zemljevida (SLGZ) [125–127]. Najpogostejše uporabljeni senzori za zaznavanje okolja so: ultrazvočni merilnik oddaljenosti, laserski merilnik oddaljenosti (linijski ali 3D) in kamera (aktivna ali pasivna).

Zemljevid mora biti v primerni obliki, da ga lahko uporabimo za lokalizacijo in načrtovanje poti. Zemljevide delimo v metrične in topološke (grafi). Za zapis metričnega zemljevida se uporabljajo mreže zasedenosti (angl. „*occupancy grid-map*”) in parametrični zemljevidi [42, 107, 123, 126, 128]. Pri uporabi kamere za gradnjo zemljevida je prostorsko potratno, če zemljevid zgradimo iz slik okolice. Bolj primerno je, da slike opišemo z značilkami in tudi zemljevid predstavimo z oblakom značilk in višjenivojskim grafom. Taka predstavitev prostora ni le prostorsko učinkovita, temveč omogoča tudi hitro obdelavo. Vendar je tudi s takšno predstavitvijo zemljevida problem lokalizacije zahtevna naloga.

Pri gradnji zemljevida in lokalizaciji se uporabljajo algoritmi, ki temeljijo na modelu porazdelitve verjetnosti [42, 128]: Bayesov filter [77, 129], Kalmanov filter, razširjeni Kalmanov filter [123], nepristranski Kalmanov filter, filter delcev [77], skriti Markov model. Pri SLGZ je še posebej pereč *problem zaključene poti*, kjer zaradi lezenja pride do nezaželenega prekrivanja zemljevida. Problem SLGZ se še dodatno zaostri v dinamičnih okoljih s pomičnimi ovirami. Omeniti velja še *problem ugrabitve*³ mobilnega sistema.

Sisteme za lokalizacijo delimo na sisteme za globalno določanje položaja in sisteme za lokalno določanje položaja. O globalnem sistemu za določanje položaja govorimo, ko imamo na voljo zunanji sistem, ki nam omogoča določanje našega položaja. Dandanes imamo povsod na svetu na voljo vsaj eden sistem izmed

³Ugrabitev mobilnega sistema je problem, ki nastane, ko mobilni sistem prenesemo na poljubno lokacijo.

globalnih sistemov za satelitsko določanje položaja (ameriški sistem GPS, ruski sistem GLONASS, kitajski sistem Beidou in projekt Compass, evropski projekt Galileo idr.). Satelitski sistemi za določanje položaja ne delujejo dobro znotraj stavb. Za delovanje znotraj stavb obstajajo posebni sistemi za lokalizacijo, vendar takšni sistemi niso povsod na voljo in jih običajno tudi ni mogoče vgraditi v poljubno okolje. O lokalnem določanju položaja govorimo, ko določimo svoj položaj le z uporabo notranjih senzorjev, tj. senzorjev, ki se nahajajo na mobilnem sistemu, pri čemer se ne zanašamo na noben drug sistem, ki ni na mobilnem sistemu. Ker je razpoložljivost in natančnost zunanjih sistemov za določanje položaja omejena, se avtonomni mobilni sistemi nikoli ne morejo povsem zanašati na zunanje sisteme za lokalizacijo. Uporaba notranjih senzorjev za zaznavanje okolja je ključna za izvedbo res avtonomnih mobilnih sistemov. Razvoj polprevodniške tehnologije je omogočil, da je dandanes možno sprejemnik signalov za globalno določanje položaja vgraditi v tako rekoč katerokoli napravo, saj je celoten sprejemnik izveden v enem samem čipu izredno majhne velikosti (manjši od velikosti zrna) po izredno dostopnih cenah (v rangi enega evra).

V večini primerov zahteva izvedba vodenja robotskega mehanizma poznavanje (inverznega) kinematičnega modela robota [130]. Vodenje robota nato poteka tako, da glede na želeno referenčno lego vrha robota, na podlagi inverznega kinematičnega modela izračunamo referenčne kote/položaje v sklepih robota in nato reguliramo le kote oz. položaje v sklepih. Tak princip deluje v primeru, ko imamo na voljo natančen model robota in ima robot veliko ponovljivost. V mnogih robotiziranih nalogah zahtevamo dvig in prenos objektov, pri čimer lahko pride zaradi teže bremena ali drugih motenj do povesa segmentov, tako da vodenje robota le na podlagi inverznega kinematičnega modela ni več možno. Problem se v tem primeru rešuje z neposredno regulacijo vrha robota, za kar pa moramo znati izmeriti pogrešek med trenutno in želeno lego robota. Nadvse primeren način za reševanje le-te naloge je s pomočjo slike [3–5, 129, 131], saj lahko kamero namestimo skoraj na vsako robotsko roko v skoraj kateremkoli okolju. Med drugim je bila metodologija vodenja na osnovi slike uporabljena v aplikacijah, kot so obiranje sadja [34], kirurške operacije [33] itd. Princip vodenja na osnovi slike je bil uporabljen tudi skupaj z regulacijo sile za vstavljanje objektov v ozke odprtine [132].

Pri vodenju robotske roke predpostavljamo, da nimamo natančno znanega

kinematičnega modela roke. Zato v prvi fazi s postopkom kalibracije ocenimo parametre modela robotske roke. Kalibracijo izvedemo tako, da z robotsko roko naredimo nekaj določenih enostavnih gibov in na sliki spremljamo odziv značilk na premik. Na podlagi meritev nato določimo osnovne parametre modela robotske roke, ki povezuje gibanje robotske roke s spreminjanjem značilk na sliki — določimo t. i. interakcijsko matriko [3]. Model robotske roke nato uporabimo pri načrtovanju regulatorja za premik robotske roke. Regulacijski zakon zasnujemo tako, da regulacija vrha izniči pogrešek v ustaljenem stanju tudi, če se dejanski model razlikuje od ocenjenega modela robotske roke. Pri načrtovanju regulatorja za vodenje robotske roke na osnovi slike, se poslužujemo podobnih principov vodenja kot za vodenje mobilnih sistemov. V kolikor je robotska roka nameščena na mobilni sistem, lahko na lego vrha robotske roke vplivamo ne le s premikanjem sklepov roke, temveč tudi s premikanjem mobilnega sistema.

3 Zaznavanje s kamero

Predmeti v prostoru svetlobo bodisi oddajajo bodisi jo odbijajo. Predmete, ki svetlobo oddajajo, imenujemo izvori svetlobe. Izvori svetlobe so sonce, ogenj, strela, električna žarnica itd. S kamero lahko poiščemo predmete v prostoru, tako da zaznamo svetlobo, ki nastane na zaslonu kamere. Na zaslonu je običajno nameščeno tipalo, ki omogoča detekcijo količine prejete svetlobe in pretvorbo te informacije v električni oz. digitalni signal. Kako nastane slika na zaslonu, lahko opišemo z različnimi matematičnimi modeli. Kateri zapis izberemo, je odvisno od namena uporabe modela. V nadaljevanju je predstavljena perspektivna transformacija, ki na področju strojnega vida velja za eno izmed pomembnejših preslikav in v veliki večini primerov predstavlja zadovoljiv model kamere. Poleg perspektivnega modela kamere je podanih nekaj posebnih lastnosti te preslikave, ki jih lahko koristno uporabimo za izvedbo kalibracije kamere.

3.1 Perspektivna projekcija

Nastanek slike na zaslonu lahko opišemo z geometrijskim modelom kamere. Obstaja mnogo matematičnih opisov kamere glede na tip kamere (običajna perspektivna kamera, kamera s polnim vidnim poljem, globinska kamera itd.) in glede na natančnost modela, ki lahko upošteva mnoge prispevke k nastanku slike — med slednje največji del običajno prispevajo popačitve leč. To delo obravnava vodenje na osnovi slike z uporabo običajne kamere. Pri obravnavi smo se omejili na uporabo dokaj enostavnega modela kamere, to je t. i. *luknjičarke* oz. *kamere na luknjico* oz. *camera obscura* (angl. „*pinhole camera*”) (slika 3.1), ki ga



Slika 3.1: (a) *Camera obscura* v človeški velikosti postavljena v pristanišču mesta Cape Town in (b) posnetek slike, ki je nastala na stenah znotraj naprave.

matematično opišemo na sledeč način¹:

$$\lambda {}^P \underline{\mathbf{p}} = \mathbf{S} {}^C \mathbf{p} = \mathbf{S} ({}^C \mathbf{R}_W {}^W \mathbf{p} + {}^C \mathbf{t}_W) . \quad (3.1)$$

kjer je λ utežni faktor, ${}^P \underline{\mathbf{p}}^T = [{}^P x \ {}^P y \ 1]$ slika točke ${}^W \mathbf{p}^T = [{}^W x \ {}^W y \ {}^W z]$ v prostoru (${}^C \mathbf{p}^T = [{}^C x \ {}^C y \ {}^C z]$ je položaj točke v koordinatnem sistemu kamere), matrika $\mathbf{S} \in \mathbb{R}^3 \times \mathbb{R}^3$ vsebuje notranje parametre kamere, rotacijska matrika ${}^C \mathbf{R}_W \in \text{SO}(3)$ in translacijski vektor ${}^C \mathbf{t}_W \in \mathbb{R}^3$ pa opisujeta lego kamere v svetovnem koordinatnem sistemu (zunanji parametri kamere). Običajno lahko predpostavimo, da ima matrika notranjih parametrov naslednjo obliko:

$$\mathbf{S} = \begin{bmatrix} \alpha & \gamma & t_x \\ 0 & \beta & t_y \\ 0 & 0 & 1 \end{bmatrix} , \quad (3.2)$$

Matriko $\mathbf{S}$ sestavljajo naslednji notranji parametri: faktorja skaliranja α in β v horizontalni in vertikalni smeri, središče optične osi (t_x, t_y) ; in strižni faktor γ . Ker velja $\lambda = \lambda({}^W \mathbf{p})$, je model (3.1) nelinearen². Slednja lastnost je bolj razvidna iz neposrednega zapisa:

$${}^P x = \frac{[1 \ 0 \ 0] \mathbf{S} ({}^C \mathbf{R}_W {}^W \mathbf{p} + {}^C \mathbf{t}_W)}{[0 \ 0 \ 1] \mathbf{S} ({}^C \mathbf{R}_W {}^W \mathbf{p} + {}^C \mathbf{t}_W)} \quad \text{in} \quad {}^P y = \frac{[0 \ 1 \ 0] \mathbf{S} ({}^C \mathbf{R}_W {}^W \mathbf{p} + {}^C \mathbf{t}_W)}{[0 \ 0 \ 1] \mathbf{S} ({}^C \mathbf{R}_W {}^W \mathbf{p} + {}^C \mathbf{t}_W)} . \quad (3.3)$$

¹V tem delu uporabljamo za označevanje svetovnega koordinatnega sistema oznako W , oznako C za koordinatni sistem kamere in oznako P za koordinatni sistem slike.

²Podrobnejša izpeljava matematičnega modela kamere je podana v dodatku B.

Ta model kamere je uporaben tudi v primeru, ko imamo opravka s popačitvami leče, le da je v tem primeru obvezno izvesti primeren kalibracijski postopek, ki odstrani popačitev, tako da lahko sliko obravnavamo, kot da je nastala po modelu idealne kamere.

V aplikacijah strojnega vida je pomembna preslikava, ki povezuje točke na dveh slikah, ki pripadata isti točki v prostoru. To preslikavo opišemo bodisi s *fundamentalno* bodisi z *esencialno* matriko.

V tem delu je v nadaljevanju kar nekaj pozornosti namenjeno posebnemu primeru preslikave, ki jo lahko uporabimo, kadar opazujemo ravne ploskve v prostoru oz. ko se kamera vrti le okoli svojega gorišča. Enačba (3.1) se v tem primeru nekoliko poenostavi. Če predpostavimo, da se opazovana ploskev v prostoru nahaja v ravnini, ki jo napenjata vektorja x in y ($z = 0$) pridemo po nekaj matematične telovadbe do preslikave v naslednji obliki:

$$\lambda {}^P \underline{p} = \mathbf{H} {}^W \underline{p}, \quad (3.4)$$

kjer je matrika $\mathbf{H}^T = [\mathbf{h}_1 \ \mathbf{h}_2 \ \mathbf{h}_3] \in \mathbb{R}^3 \times \mathbb{R}^3$ bolj znana pod imenom *homografija*. Pri tem smo definirali točko na ravnini xy kot ${}^W \underline{p}^T = [{}^W x \ {}^W y \ 1]$. Velja: $\lambda = \mathbf{h}_3^T {}^W \underline{p}$. Preslikavo (3.4) lahko zopet zapišemo v še nekoliko drugačni obliki, iz katere je takoj razvidno, da gre za nelinearno preslikavo:

$${}_P x = \frac{\mathbf{h}_1^T {}^W \underline{p}}{\mathbf{h}_3^T {}^W \underline{p}} \quad \text{in} \quad {}_P y = \frac{\mathbf{h}_2^T {}^W \underline{p}}{\mathbf{h}_3^T {}^W \underline{p}}. \quad (3.5)$$

V določenih pogojih lahko uporabimo posebno postavitve, ki omogoča poenostavljen matematični model. Take postavitve so znane kot *kanonične*. Primer take kanonične postavitve je, ko je ravnina slike vzporedna z ravnino opazovane ploskve. V tem primeru zavzame homografija posebno (enostavnejšo) obliko:

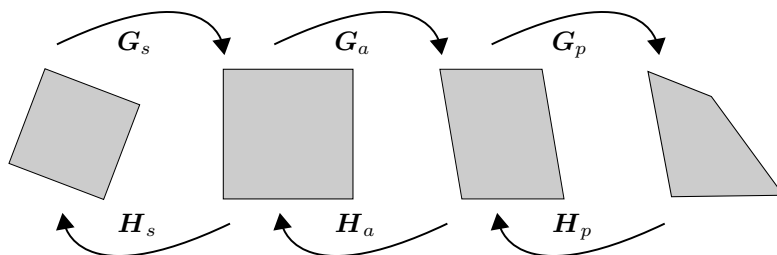
$$\mathbf{H} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.6)$$

kar pomeni, da je točka na ploskvi le skalirana in premaknjena na slikovno ravnino, torej: ${}_P x = s_x {}^W x + t_x$ in ${}_P y = s_y {}^W y + t_y$; kjer sta parametra (s_x, t_x) skalirni in translacijski faktor v horizontalni smeri in parametra (s_y, t_y) skalirni in translacijski faktor v vertikalni smeri. Takšna postavitve je bila uporabljena v primeru male lige robotskega nogometa [72].

Zaradi enostavnosti uvedimo novo oznako za inverzno homografijo $\mathbf{G}$: $\mathbf{H} = \mathbf{G}^{-1}$. Homografijo $\mathbf{H}$ lahko zapišemo kot produkt treh elementarnih transformacij: podobnostne transformacije, afine transformacije in čiste projekcijske transformacije:

$$\mathbf{H} = \underbrace{\begin{bmatrix} s \cos \varphi & -s \sin \varphi & t_x \\ s \sin \varphi & s \cos \varphi & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{H}_s} \underbrace{\begin{bmatrix} \frac{1}{b} & -\frac{a}{b} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{H}_a} \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ l_1 & l_2 & l_3 \end{bmatrix}}_{\mathbf{H}_p}. \quad (3.7)$$

Transformacijska matrika $\mathbf{H}_p$ preslika točke iz prostora slike v afini prostor, transformacija $\mathbf{H}_a$ iz afinega prostora v metrični prostor in matrika $\mathbf{H}_s$ iz metričnega v Evklidski prostor. Vpliv, ki ga imajo te transformacije na kvadrat je razviden s slike 3.2. V podobnostni matriki predstavlja element s skalirni faktor, urejen par (t_x, t_y) translacijo in kot φ desnosučno rotacijo v ravnini (okoli osi z). Parametra a in b v preslikavi iz afinega v metrični prostor določata skrček in strig. Zadnja vrstica v čisti projekcijski matriki $\mathbf{H}_p$ vsebuje parametre premice horizonta opazovane ravnine $\mathbf{l}_{p\infty}^T = [l_1 \ l_2 \ l_3]$.



Slika 3.2: Vplivi treh osnovnih preslikav na obliko kvadrata, od leve proti desni: podobnostna, afina in čista perspektivična preslikava.

Za vsako točko $\mathbf{p} \in \mathbb{R}^2$, ki leži na premici $\mathbf{l} \in \mathbb{R}^3$ velja $\mathbf{l}^T \mathbf{p} = 0$. Presečišče $\mathbf{x}$ med premicama $\mathbf{l}_1$ in $\mathbf{l}_2$ podaja vektorski produkt $\underline{\mathbf{p}} = \mathbf{l}_1 \times \mathbf{l}_2$. Dualno, je premica $\mathbf{l}$, ki poteka skozi točki $\mathbf{p}_1$ in $\mathbf{p}_2$, podana z vektorskim produktom $\mathbf{l} = \underline{\mathbf{p}}_1 \times \underline{\mathbf{p}}_2$.

Premice so invariantne na perspektivno transformacijo, kar pomeni, da se premice preslikajo v premice: ${}^P\mathbf{l} = \mathbf{H}^{-T} {}^W\mathbf{l}$. Ugotovili smo, da so tako točke kot premice invariantne na perspektivno transformacijo. Velja pa še bolj splošna

lastnost, da so vse stožnice invariantne na perspektivno transformacijo. Enačbo splošne stožnice v implicitni obliki podamo na sledeč način:

$$A {}^Wx^2 + B {}^Wx {}^Wy + C {}^Wy^2 + D {}^Wx + E {}^Wy + F = 0, \quad (3.8)$$

ki jo lahko predstavimo tudi v kompaktni matrični obliki:

$${}^W\underline{\mathbf{p}}^T {}^W\mathbf{C} {}^W\underline{\mathbf{p}} = 0, \quad {}^W\mathbf{C} = \begin{bmatrix} A & B/2 & D/2 \\ B/2 & C & E/2 \\ D/2 & E/2 & F \end{bmatrix}, \quad (3.9)$$

kjer so vsi parametri stožnice zbrani v simetrični matriki ${}^W\mathbf{C}$. Če nadomestimo vektor ${}^W\underline{\mathbf{p}}$ z enačbo (3.4), dobimo preslikavo stožnice na sliko ${}^P\underline{\mathbf{p}}^T {}^P\mathbf{C} {}^P\underline{\mathbf{p}} = 0$:

$${}^P\mathbf{C} = \mathbf{H}^T {}^W\mathbf{C} \mathbf{H}. \quad (3.10)$$

Za nas je zelo zanimiva preslikava med kotom v ravnini v svetovnem koordinatnem sistemu ${}^W\varphi$ in njegovo sliko v koordinatnem sistemu slike ${}^P\varphi$. Do zelene relacije lahko pridemo z odvajanjem enačbe (3.4). Če upoštevamo $\mathbf{h}_3^T \mathbf{H} {}^P\underline{\mathbf{p}} = 1$, lahko z nekoliko matematične telovadbe pridemo do izraza:

$$\tan {}^W\varphi = \frac{(h_{21} - {}^Py h_{31}) \cos {}^P\varphi - (h_{11} - {}^Px h_{31}) \sin {}^P\varphi}{(h_{12} - {}^Px h_{32}) \sin {}^P\varphi - (h_{22} - {}^Py h_{32}) \cos {}^P\varphi}. \quad (3.11)$$

Naj opozorimo, da je enačba (3.11) odvisna od koordinate ${}^P\underline{\mathbf{p}}$ na sliki, kjer je izhodišče kota. In še, kot ${}^W\varphi$ je podan v referenčnem koordinatnem sistemu ravnine in kot ${}^W\varphi$ v referenčnem koordinatnem sistemu slike.

Poglejmo si še, kako se določena okolica okrog poljubne točke na ravnini preslika preko homografije v okolico slike te točke na sliki. Označimo s $\sigma > 0$ velikost okolice (polmer kroga). Okolico poljubne točke ${}^W\underline{\mathbf{p}}$ opišemo s krogom:

$${}^W\mathbf{C} ({}^W\underline{\mathbf{p}}, \sigma) = \begin{bmatrix} 1 & 0 & -{}^Wx \\ 0 & 1 & -{}^Wy \\ -{}^Wx & -{}^Wy & {}^Wx^2 + {}^Wy^2 - \sigma^2 \end{bmatrix}. \quad (3.12)$$

Okolico na sliki določimo s pomočjo enačb (3.4) in (3.10):

$${}^P\mathbf{C} ({}^P\underline{\mathbf{p}}, \sigma) = \mathbf{H}^T {}^W\mathbf{C} (\lambda \mathbf{H}^{-1} {}^P\underline{\mathbf{p}}, \sigma) \mathbf{H}. \quad (3.13)$$

Perspektivna transformacija je nelinearna preslikava. V kolikor imamo opravka s perspektivno preslikavo med dvema ravninama, lahko le-to opišemo s homografijo, ki je tudi nelinearna preslikava. Ugotovili smo še, da se splošna stožnica preko homografije preslika nazaj v stožnico.

3.2 Geometrija več pogledov

Z geometrijo več pogledov opišemo preslikavo med kamerami, ki so bodisi prostorsko porazdeljene po prostoru bodisi so časovno porazdeljene bodisi so tako časovno kot prostorsko porazdeljene. Obravnavo bomo na tem mestu omejili na uporabo kamer z enakimi oz. nespremenljivimi notranjimi elementi, kar pomeni da je matrika $\mathbf{S}$ konstantna oz. enaka za vse obravnavane kamere. Brez izgube na splošnosti predpostavimo, da imamo dve prostorsko porazdeljeni kameri. Povezavo med točkami med koordinatnima sistemoma kamer C_1 in C_2 podaja povezava:

$${}^{C_2}\mathbf{p} = {}^{C_2}\mathbf{R}_{C_1} {}^{C_1}\mathbf{p} + {}^{C_2}\mathbf{t}_{C_1}, \quad (3.14)$$

kjer matrika ${}^{C_2}\mathbf{R}_{C_1}$ podaja rotacijo med kamerama in vektor ${}^{C_2}\mathbf{t}_{C_1}$ podaja premik med izhodiščema kamer. Ob upoštevanju perspektivne transformacije (3.1), pridemo z nekaj matematične manipulacije:

$${}^{C_2}\mathbf{p} = {}^{C_2}\mathbf{R}_{C_1} {}^{C_1}\mathbf{p} + {}^{C_2}\mathbf{t}_{C_1}, \quad (3.15)$$

$$[{}^{C_2}\mathbf{t}_{C_1}]_{\times} {}^{C_2}\mathbf{p} = [{}^{C_2}\mathbf{t}_{C_1}]_{\times} {}^{C_2}\mathbf{R}_{C_1} {}^{C_1}\mathbf{p} + [{}^{C_2}\mathbf{t}_{C_1}]_{\times} {}^{C_2}\mathbf{t}_{C_1}, \quad (3.16)$$

$$\lambda_2 [{}^{C_2}\mathbf{t}_{C_1}]_{\times} \mathbf{S}^{-1} {}^{P_2}\underline{\mathbf{p}} = \lambda_1 [{}^{C_2}\mathbf{t}_{C_1}]_{\times} {}^{C_2}\mathbf{R}_{C_1} \mathbf{S}^{-1} {}^{P_1}\underline{\mathbf{p}}, \quad (3.17)$$

$$0 = \lambda_1 {}^{P_2}\underline{\mathbf{p}}^T \mathbf{S}^{-T} [{}^{C_2}\mathbf{t}_{C_1}]_{\times} {}^{C_2}\mathbf{R}_{C_1} \mathbf{S}^{-1} {}^{P_1}\underline{\mathbf{p}}; \quad (3.18)$$

do izraza:

$${}^{P_2}\underline{\mathbf{p}}^T {}^{P_2}\mathbf{F}_{P_1} {}^{P_1}\underline{\mathbf{p}} = 0, \quad {}^{P_2}\mathbf{F}_{P_1} = \mathbf{S}^{-T} [{}^{C_2}\mathbf{t}_{C_1}]_{\times} {}^{C_2}\mathbf{R}_{C_1} \mathbf{S}^{-1}. \quad (3.19)$$

Matrika ${}^{P_2}\mathbf{F}_{P_1}$ v izrazu (3.19) je bolj znana kot *fundamentalna matrika*. V primeru, da je matrika notranjih parametrov $\mathbf{S}$ znana, lahko popačitev zaradi perspektivne transformacije odpravimo, tako da upoštevamo ${}^P\underline{\mathbf{p}} = \mathbf{S} {}^C\underline{\mathbf{p}}$, kar nas privede do izraza:

$${}^{C_2}\underline{\mathbf{p}}^T {}^{C_2}\mathbf{E}_{C_1} {}^{C_1}\underline{\mathbf{p}} = 0, \quad {}^{C_2}\mathbf{E}_{C_1} = [{}^{C_2}\mathbf{t}_{C_1}]_{\times} {}^{C_2}\mathbf{R}_{C_1}, \quad (3.20)$$

kjer je ${}^{C_2}\mathbf{E}_{C_1}$ znana kot *esencialna matrika*. Fundamentalna in esencialna matrika sta izredno pomembni, saj podajata omejitve med točkami med dvema kamerama. S tema matrikama je za vsako točko na prvi sliki določena premica, na kateri leži točka na drugi sliki, ki je slika iste točke v prostoru, kot je točka na prvi sliki. To t. i. *epipolarno omejitev* lahko koristno uporabimo pri določevanju

parov točk med slikama, saj lahko ob poznavanju fundamentalne oz. esencialne matrike omejimo območje iskanja parov v dveh dimenzijah le na točno določene premice. V primeru, ko sta kameri postavljeni tako, da njuni slikovni ravnini ležita v isti ravnini, se iskanje parov še poenostavi. V tem primeru postanejo vse epipolarne premice vzporedne. Ko so epipolarne premice še vzporedne s slikovnimi vrsticami oz. stolpci, je iskanje še preprostejše, saj vsaki točki na prvi sliki pripada primerno zamaknjena vrstica ali stolpec na drugi sliki. V takšnih primeri govorimo o *kanoničnih postavitvah*.

Enačbi (3.19) in (3.20) veljata za vse točke, ki pripadajo isti točki kjerkoli v prostoru. V nadaljevanju je obravnavan poseben primer, ko imamo opravka le s točkami, ki ležijo v prostoru na poljubni ravnini. V tem primeru lahko izpeljemo vrsto relacij, ki se izkažejo za uporabne pri načrtovanju vodenja na osnovi slike.

3.3 Kalibracija kamere

Vsaka obširnejša knjiga o strojnem vidu vsebuje poglavje o kalibraciji kamere (npr. [38]). Problem kalibracije kamere je dobro raziskano področje. Kvaliteta izvedene kalibracije kamere običajno močno vpliva na končno kvaliteto rešitve vodenja na osnovi slike, zato si bomo v tem poglavju ogledali nekaj znanih pristopov za kalibracijsko kamere in pokazali tudi predlagano metodo za kalibracijo sistema s kamero.

Konvencionalni pristopi vodenja na osnovi slike zahtevajo posebno pripravo delovnega okolja, običajno z vnosom posebnega objekta v okolje le za namen kalibracije, na primer šahovnice [67, 70, 133] ali okroglega objekta [134] ali kakšnih drugih objektov v obliki stožnic [135–139] ali drugo. Način kalibracije je odvisen od izbranega modela kamere. Obstaja mnogo modelov kamere, ki opisujejo različne lastnosti preslikave slike z različnimi natančnostmi. Glede na zahteve moramo izbrati primeren model kamere, ki opisuje dejansko kamero z zadovoljivo natančnostjo. V mnogih situacijah zadostuje uporaba preprostega modela t. i. kamere luknjičarke. Zaradi popačitev leč lahko takšen model ni ustrezen, zato moramo uporabiti modele, ki upoštevajo tudi popačitve leč [65]. Med geometrijskimi popačitvami leč izstopata predvsem radialna in tangencialna distorzija. V model kamere vključimo tudi določene lastnosti slikovnega senzorja,

predvsem obliko na svetlobo občutljivih pik na slikovnem senzorju. V modelu kamere so običajno zajeti t. i. notranji parametri kamere kot tudi t. i. zunanji parametri kamere, ki določajo lego kamere v prostoru.

S postopkom kalibracije želimo običajno določiti tako notranje kot zunanje parametre kamere. Vendar ni vedno tako. V določenih situacijah nam zadošča poznavanje preslikave med slikovno ravnino in točkami v prostoru, brez poznavanja posameznih notranjih in zunanjih parametrov kamere. Klasični pristop kalibracije [67, 69] temelji na oceni modela kamere na podlagi množice nekoplanarnih točk v prostoru in množice pripadajočih parov točk na sliki. Model kamere je nato ocenjen z uporabo metode najmanjših kvadratov oz. ene izmed optimizacijskih metod. Metode so običajno razdeljene v dva koraka, kjer v prvem koraku določimo celotno preslikavo, nato pa v drugem koraku ocenjeno preslikavo razstavimo na posamezne elemente, tj. notranje in zunanje parametre kamere. Delo [70] predstavlja metodo kalibracije, kjer je avtor uporabil več množic koplanarnih točk, natančneje več pogledov šahovnice iz različnih zornih kotov. Metoda najprej oceni posamezne preslikave (homografije) med različnimi ravninami šahovnice in ravnino slike, nato pa na podlagi ocenjenih homografij določi parametre modela kamere. V pristopu je predstavljen tudi postopek, ki upošteva radialno distorzijo leče.

3.3.1 Kalibracija na podlagi parov točk

Za namene ocenitve elementov homografije, zapišemo enačbo (3.5) v nekoliko spremenjeni obliki:

$$\begin{bmatrix} {}^w\mathbf{p}^T & \mathbf{0}^T & -{}^P_x & {}^w\mathbf{p}^T \\ \mathbf{0}^T & {}^w\mathbf{p}^T & -{}^P_y & {}^w\mathbf{p}^T \end{bmatrix} \mathbf{h} = \mathbf{0}, \quad (3.21)$$

kjer smo vse elemente homografije zbrali v vektorju $\mathbf{h}^T = [\mathbf{h}_1^T \ \mathbf{h}_2^T \ \mathbf{h}_3^T]$. V kolikor imamo na voljo N parov točk, lahko zložimo N enačb (3.21) skupaj v linearni sistem oblike:

$$\Psi \mathbf{h} = \mathbf{0}, \quad (3.22)$$

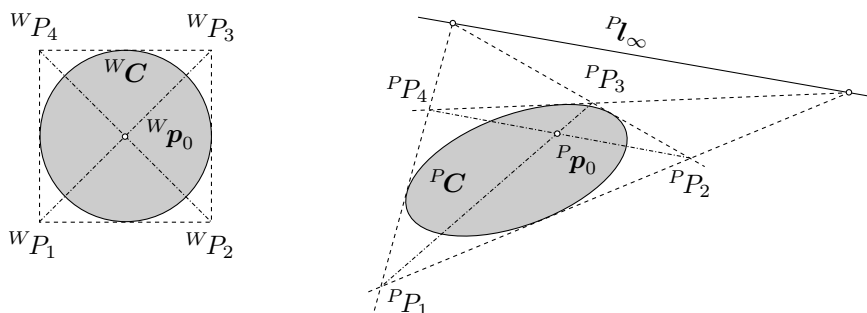
kjer je $\Psi \in \mathbb{R}^{2N} \times \mathbb{R}^9$ in je $\mathbf{0}$ ničelni vektor dimenzije $2N$. Kalibracijske točke ne smejo biti kolinearne, sicer je sistem slabo pogojen. Ker je rešitev sistema (3.22) določena z eno prosto utežjo, moramo pri reševanju vpeljati dodatne pogoje.

Enostavna možnost je, da enega izmed elementov v vektorju $\mathbf{h}$ postavimo na vrednost ena (npr. $h_{33} = 1$). Vendar takšna izbira ni vedno najbolj primerna [70]. Običajno se uporabi zahteva $\mathbf{h}^T \mathbf{h} = 1$. Rešitev sistema je v tem primeru lastni vektor matrike $\Psi^T \Psi$, ki pripada najmanjši lastni vrednosti. Direktna linearna transformacija za ravnino je podrobneje opisana v dodatku C. Vrednosti v sistemu (3.22) so podane v različnih merskih enotah, kar običajno pomeni, da je definicijsko območje vrednosti razpršeno po mnogih različnih dekadah (npr. določene veličine so podane v metrih, določene v slikovnih koordinatah, določene so podane z izpeljanimi enotami). To dejstvo ima slab vpliv na numerično stabilnost rešitve. V [71] je bila zato predlagana uporaba *središčnega raztega* (glej dodatek D). Pred samo izvedbo ocene parametrov $\mathbf{h}$, vhodne podatke Ψ preslikamo z ustrezno preslikavo, ki poskrbi za primerno normiranje podatkov. Nato na preslikanih podatkih poiščemo najboljšo oceno parametrov v smislu najmanjših kvadratov, ki jo v zadnjem koraku s pomočjo uporabljene transformacije v prvem koraku preslikamo v končno oceno. Dobljena ocena je določena na podlagi minimizacije algebrajske napake, a pri ocenjevanju parametrov le-ta ne daje vedno najboljših rezultatov. Pogosto se zato namesto algebrajske napake uporabi geometrijska napaka, ki glede na znano znanje o sistemu bolj smiselno opiše optimizacijski problem. Rešitev nato običajno poiščemo z uporabo nelinearnih optimizacijskih metod [70]. Omenjeni postopek ocenitve parametrov homografije je mogoč, v kolikor je popačitev zaradi distorzije leč zanemarljiva. Sicer moramo optimizacijski problem razširiti tako, da poleg ocenjevanja parametrov homografije ocenimo tudi parametre leče. Glede na zahtevano natančnost in tip distorzije moramo izbrati primeren model leč [65]. Na oceno parametrov imata običajno največji vpliv radialna in tangencialna distorzija leče. Pri tem je uporaba nelinearnih optimizacijskih metod neizogibna.

3.3.2 Kalibracija s pomočjo stožnic

Pri perspektivni preslikavi splošne stožnice se srečamo z določenimi posebnimi lastnostmi, ki so predstavljene v tem poglavju in uporabljene tudi v nadaljevanju. Vedno, kadar imamo opravka z dvema poljubnima stožnicama v poljubni medsebojni legi, velja, da se ti dve stožnici sekata v štirih realnih ali kompleksnih točkah. Dve poljubni koplanarni krožnici se vedno sekata v dveh kompleksnih

točkah v neskončnosti, ${}^w\mathbf{p}_{\{a,b\}}^T = [1 \pm i \ 0]$, ki ju imenujemo *krožni točki* (angl. „circular points”). Ti dve krožni točki ležita na t. i. *premici v neskončnosti* (angl. „line at infinity”) ${}^w\mathbf{l}_\infty^T = [0 \ 0 \ 1]$ in se preko perspektivne projekcije preslikata v sliki krožnih točk ${}^P\mathbf{p}_a$ in ${}^P\mathbf{p}_b$, ki ležita na bežnici ${}^P\mathbf{l}_\infty$.



Slika 3.3: Preslikava krožnice in vzporenih premic na ravnini (levo) preko perspektivne transformacije na slikovno ravnino (desno).

Omenjene lastnosti pri perspektivni preslikavi stožnic lahko uporabimo za določitev t. i. podobnostne homografije $\mathbf{H}_a\mathbf{H}_p$ na podlagi slike krožnice ${}^P\mathbf{C}$ in slike središča krožnice ${}^P\mathbf{p}_0$. Čista perspektivna homografija $\mathbf{H}_p$ je enolično določena z bežnico ${}^P\mathbf{l}_\infty^T = [l_1 \ l_2 \ l_3]$ (glej (3.7)), ki je slika premice v neskončnosti na svetovni ravnini. Znano je, da se vse vzporedne premice v svetovni ravnini sekajo v eni točki na slikovni ravnini — v t. i. bežišču —, kar je prikazano s črtkanimi premicami na sliki 3.3. Snopi vzporednih premic — pri čemer premice med poljubnima snopoma niso vzporedne — s skupne ravnine, določajo natanko toliko bežišč kolikor je snopov, pri čemer vsa bežišča ležijo na isti premici, na t. i. *bežnici* ${}^P\mathbf{l}_\infty$.

V kolikor je celotni krog na ravnini v prostoru pred slikovno ravnino, se krog v splošnem preslika v elipso na sliki. Velja, da središče elipse na sliki, ki je slika krožnice v prostoru, ne sovпада s sliko središča tega kroga, kar je razvidno tudi s slike 3.3. Slika kroga (elipsa) ${}^P\mathbf{C}$, slika središča kroga ${}^P\mathbf{p}_0$ in bežnica ${}^P\mathbf{l}_\infty$ so v t. i. *polarni zvezi*:

$${}^P\mathbf{l}_\infty = {}^P\mathbf{C} {}^P\mathbf{p}_0. \quad (3.23)$$

S pomočjo zveze (3.23) lahko neposredno določimo enačbo bežnice (slika 3.3).

Bežnica ${}^P\mathbf{l}_\infty$ seka sliko kroga (elipso) ${}^P\mathbf{C}$ v dveh navideznih točkah (slikah krožnih točk) ${}^P\mathbf{p}_a$ in ${}^P\mathbf{p}_b$. Enostavno lahko pokažemo, da sta sliki krožnih točk invariantni na podobnostno preslikavo. Zatorej lahko na podlagi slik krožnih točk analitično določimo vse parametre podobnostne homografije $\mathbf{H}_a\mathbf{H}_p$, saj velja, da se krožni točki ${}^W\mathbf{p}_{\{a,b\}}$ preslikata v sliki krožnih točk ${}^P\mathbf{p}_{\{a,b\}} = [b \pm ia, \pm i, -\frac{l_1 b}{l_3} \mp \frac{l_1 a + l_2}{l_3}]$.

3.4 Kalibracija iz gibanja

Postopek kalibracije kamere opisan v poglavju 3.3 zahteva poznavanje položajev točk na ravnini v svetovnem koordinatnem sistemu in položaje pripadajočih slik teh točk na slikovni ravnini oz. v koordinatnem sistemu slike. V tem poglavju so obravnavi kalibracijski postopki, ki omogočajo ocenitev podobnostne homografije na podlagi točke, ki se s konstantno hitrostjo suče okoli poljubne osi v prostoru in se torej giblje po krožnici.

Predpostavimo, da kamera opazuje točko, ki se v ravnini vrti okoli nepremične osi. Gibanje točke (${}^Wx, {}^Wy$) v svetovni ravnini lahko opišemo s sledečim matematičnim modelom:

$$\begin{aligned} {}^Wx(t) &= \int_0^t v(\tau) \cos({}^W\varphi(\tau)) d\tau + {}^Wx(0), \\ {}^Wy(t) &= \int_0^t v(\tau) \sin({}^W\varphi(\tau)) d\tau + {}^Wy(0), \\ {}^W\varphi(t) &= \int_0^t \omega(\tau) d\tau + {}^W\varphi(0), \end{aligned} \tag{3.24}$$

kjer v in ω predstavljata tangencialno in kotno hitrost gibanja točke. Začetni pogoji določajo položaj in orientacijo (lego) izhodišča svetovnega koordinatnega sistema. Izbira začetnih pogojev ne vpliva na postopek ocenjevanja preslikave med svetovnim koordinatnim sistemom in slikovno ravnino, zato lahko začetne pogoje izberemo poljubno, saj nas zanimajo le relativni premiki glede na začetno lego.

Numerična integracija sistema enačb (3.24) vnaša lezenje, zato integracijska metoda daje relativno točne rezultate le za relativno kratke čase integracije.

Predpostavimo, da sta tangencialna in kotna hitrost konstantni neničelni vrednosti, kar pomeni, da v ustaljenem stanju dosežemo gibanje točke po krožnici. Pri uporabi konstantnih vhodnih hitrosti v in ω , se izognemo tako problemu zakasnitev kot tudi problemu numerične integracije, kot je razvidno iz rezultatov v nadaljevanju. Z izbiro $v(t) = v = \text{konst.}$ in $\omega(t) = \omega = \text{konst.}$ lahko analitično poenostavimo sistem enačb (3.24):

$$\begin{aligned} {}^Wx(t) &= \frac{v}{\omega}(\sin(\omega t + {}^W\varphi(0)) - \sin({}^W\varphi(0))) + {}^Wx(0), \\ {}^Wy(t) &= -\frac{v}{\omega}(\cos(\omega t + {}^W\varphi(0)) - \cos({}^W\varphi(0))) + {}^Wy(0). \end{aligned} \quad (3.25)$$

Predpostavimo, da so začetni pogoji izbrani, kot sledi: ${}^Wx(0) = \frac{v}{\omega}$, ${}^Wy(0) = 0$ in ${}^W\varphi(0) = \frac{\pi}{2}$. Enačbi (3.25) se sedaj še poenostavita:

$${}^Wx(t) = \frac{1}{\kappa} \cos(\omega t) \quad \text{in} \quad {}^Wy(t) = \frac{1}{\kappa} \sin(\omega t). \quad (3.26)$$

Razmerje med kotno in tangencialno hitrostjo predstavlja ukrivljenost krožne poti $\kappa = \frac{\omega}{v}$. Za dosego zadovoljive natančnosti ocene homografije morata biti hitrosti v in ω primerno izbrani, da dosežemo primerno velik radij (majhno ukrivljenost) krožne poti.

3.4.1 Direktna metoda: Ocena homografije na podlagi parov točk

Homografijo lahko ocenimo na podlagi znanih parov točk med dvema ravninama, po dokaj klasičnem postopku, kot je bilo opisano že v poglavju 3.3. V našem primeru določimo položaje točk v svetovni ravnini, tako da merimo čas in na podlagi znane tangencialne in kotne hitrosti z enačbo (3.26) določimo položaje točk v svetovnem koordinatnem sistemu. Obenem beležimo še položaje točk na slikovni ravnini, in tako pridemo do parov točk med ravninama.

Takšen postopek zahteva poznavanje oz. merjenje kotne hitrosti vrteče se točke. Točnost poznavanja kotne hitrosti močno vpliva na točnost ocenjenih položajev točk s pomočjo enačbe (3.26), saj je kotna hitrost neposredno povezana z dolžino časovne periode krožnice. Ker je časovna perioda krožnice enaka časovni periodi njene slike (elipse), lahko kotno hitrost ocenimo na podlagi časa, ki je potreben za en zaključen krog (eno zaključeno elipso). To je seveda možno le, če je vrteča se točka opravila najmanj en polni krog. Postopek zahteva tudi poznavanje

tangencialne hitrosti, a točna ocena tangencialne hitrosti ni tako pomembna, saj netočna ocena tangencialne hitrosti vpliva le na netočnost ocene radija kroženja, in torej vpliva na velikost enotskega vektorja v svetovnem koordinatnem sistemu.

3.4.2 Lourakisova metoda: Ocena homografije na podlagi koplanarnih krogov

Postopek predstavljen v [139] omogoča ocenitev homografije na podlagi dveh ali več koplanarnih krogov. Pri tem ni potrebno poznavanje parametrov teh krogov, pomembno je le, da je oblika objektov res krožna in da vsi krogi ležijo v isti ravnini. Sledi kratek povzetek algoritma. V prvi fazi postopka skozi množice posnetih točk, ki so slike $n > 1$ koplanarnih krožnic, napnemo elipse, ki se danim točkam najboljše prilagajo. Nato uporabimo dvokoračni algoritem (postavitev hipoteze in nato preverjanja le-te) za iskanje položaja preslikanega središča enega izmed krogov (recimo kroga i) na sliko. V prvem koraku napravimo hipotezo o legi slike središča i -tega kroga ${}^P\mathbf{p}_0$ nekje znotraj elipse (slike kroga). Na podlagi predpostavljenega položaja slike središča kroga izračunamo homografijo, ki preslika elipso v krog, kar storimo s pomočjo enačb predstavljenih v poglavju 3.3.2. Sledi preverjanje hipoteze, ki jo opravimo tako, da vse preostale $n - 1$ elipse preslikamo s predpostavljeno homografijo nazaj v svetovni prostor. Ker smo predpostavili, da so elipse slike koplanarnih krogov, bi se v primeru potrditve hipoteze morale vse elipse preslikati v kroge. V kolikor temu ni tako, je predpostavka napačna. Da določimo, kako dobri krogi so preslikane elipse, uvedemo mero za določanje krožne oblike preslikanih elips, npr. s primerjavo velike polosi M_j in male polosi m_j preslikanih elips:

$$E'({}^P\mathbf{p}_0) = \sum_{\substack{j=1 \\ j \neq i}}^n \left| 1 - \left(\frac{m_j}{M_j} \right)^2 \right|. \quad (3.27)$$

Več hipotez naredimo tako, da izberemo več možnih točk središča i -tega kroga ${}^P\mathbf{p}_0$ znotraj slike i -tega kroga. Homografija, ki preslika vse ostale elipse v najbolj idealne kroge (homografija pri najmanjši vrednosti cenilke E'), je iskana rešitev.

Predstavljeni algoritem je bil zasnovan za uporabo na statičnih slikah krogov. Vendar je metodo možno razširiti tudi na problem, ko sledimo točkam, ki se vrtijo v po krožnici v skupni ravnini. Tudi v takšnem primeru algoritem ne zahteva

poznavanja kotne hitrosti in časa odčitkov meritev. Ker je ocena homografije narejena na podlagi stožnic, običajno s takšnim postopkom lahko dosežemo zelo dobre rezultate. Za doseg zadovoljive točnosti in v izogib lokalnim minimumom moramo pri tem postopku uporabiti primerno visoko resolucijo področja, v katerem iščemo rešitev. Z večanjem resolucije iskalnega področja pa se povečuje tudi časovna zahtevnost algoritma.

3.4.3 Predlagana metoda: Ocena homografije iz gibanja

Predpostavimo, da opazujemo točko, ki se vrti s konstantno kotno hitrostjo okoli določene osi v svetovnem koordinatnem sistemu. Na sliki merimo položaj točke skozi čas. Na ta način dobimo množico N točk na sliki s pripadajočimi časi odčitkov, $\{({}^P\mathbf{p}_i, t_i)\}_{i=1,2,\dots,N}$. V kolikor je opazovana točka vedno pred slikovno ravnino, tvori slika množice točk kroga v svetovnem koordinatnem sistemu elipso na sliki. Predlagani postopek ocenjevanja homografije gre, kot sledi. V prvem koraku napnemo na množico točk $\{{}^P\mathbf{p}_i\}_{i=1,2,\dots,N}$ elipso ${}^P\mathbf{C}$, ki se najbolj prilega podatkom. Obstaja veliko različnih načinov za optimalno prilagajanje stožnice (elipse) na podatke [140, 141]. Ker vemo, da je slika kroga elipsa, je najbolje, da uporabimo metodo, ki upošteva to dejstvo v samem postopku (tak je npr. algoritem [142]). V kolikor so znane še koordinate slike središča kroga, lahko določimo iskano homografijo. Do tega koraka je postopek enak kot v Lourakisovem pristopu. Sedaj namesto da uporabimo več koplanarnih krogov za iskanje slike središča kroga, uporabimo v ta namen časovno vrsto $\{t_i\}_{i=1,2,\dots,N}$, kot sledi. Osnovna ideja pristopa je, da je središčni kot v svetovnem koordinatnem sistemu ${}^W\varphi$ proporcionalen času opazovanja t , saj smo predpostavili, da se opazovana točka vrti s konstantno kotno hitrostjo.

Celotni postopek ocenjevanja homografije v strnjeni obliki je tako:

1. Elipso ${}^P\mathbf{C}$ napnemo na množico točk $\{{}^P\mathbf{p}_i\}_{i=1,2,\dots,N}$, tako da se le-tej množici kar najbolj prilega.
2. Področje znotraj elipse razdelimo z mrežo.
3. Za vsako vozliščno točko z mreže naredimo:

(a) Trenutno izbrano točko označimo s ${}^P\mathbf{p}_0^T = [{}^Px_0 \ {}^Py_0]$.

- (b) Izračunamo homografijo $\mathbf{H}$, ki preslika elipso ${}^P\mathbf{C}$ v krog in točko ${}^P\mathbf{p}_0$ v središče tega kroga, kot je opisano v poglavju 3.3.2.
- (c) Izračunamo urejeno množico kotov $\{\varphi_i = \arg({}^P\mathbf{p}_i - {}^P\mathbf{p}_0)\}_{i=1,2,\dots,N}$ in jih vstavimo v enačbo (3.11), da dobimo urejeno množico kotov $\{\varphi_i^W\}_{i=1,2,\dots,N}$.
- (d) S pomočjo metode najmanjših kvadratov poiščemo parametra ω in ϕ , ki minimizirata kriterijsko funkcijo:

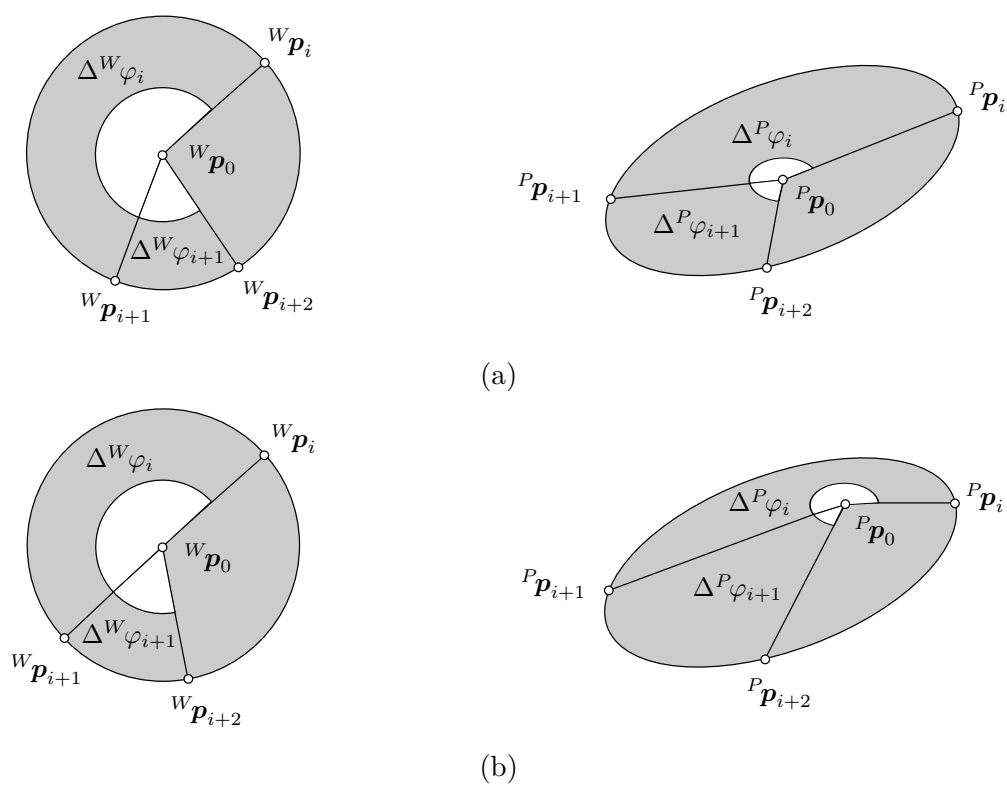
$$E({}^P\mathbf{p}_0) = \min_{\omega, \phi} \sum_{i=1}^N \varphi_i^W - \omega t_i - \phi, \quad (3.28)$$

kjer je ϕ skalar, ki določa začetni kot v svetovnem koordinatnem sistemu in ω je neznana kotna hitrost.

4. Izberemo ${}^P\mathbf{p}_0$ kot ocenjene koordinate slike središča kroga, ω kot ocenjeno kotno hitrost vrteče se točke in $\mathbf{H}$ kot oceno iskane homografije, pri kateri ima kriterijska funkcija E med vsemi točkami z mreže najmanjšo vrednost.

Dve različni hipotezi glede slike središča kroga sta prikazani na sliki 3.4. V primeru, da je postavljena hipoteza prava slika središča kroga, so središčni koti v svetovnem koordinatnem sistemu proporcionalni izmerjenemu času. Enostavno lahko pokažemo, da ima kriterijska funkcija E lokalne minimume v točkah, ki ležijo na elipsi (sledi iz znane zveze med obodnim in središčnim kotom). Da se izognemo problemu lokalnih minimumov, izberemo okoli elipse določeno prepovedano območje v katerem ne preverjamo hipotez. Za običajne postavitve (tiste, ki niso blizu singularnostim) se znotraj prepovedanega območja ne nahaja iskana rešitev. V kolikor najdemo rešitev ravno na robu prepovedanega območja, tako rešitev zavrnilo in torej pri takšni postavitvi ocenjevanje homografije na omenjen način ni možno.

Predlagana metoda zahteva opazovanje le ene točke, ki se vrti v krogu v svetovni ravnini. Lastnosti predlagane metode v primerjavi z direktno metodo (predstavljeno v poglavju 3.4.1) so: podobnostno homografijo je moč oceniti tudi, ko imamo na voljo le del kroga; med postopkom ocenjevanja se oceni tudi kotna hitrost. V primerjavi z direktno metodo je postopek dokaj računsko zahteven, saj moramo močno razdrobiti iskalno področje znotraj elipse, če želimo doseči zadovoljivo natančnost. V tem pogledu je časovna zahtevnost primerljiva z



Slika 3.4: Dve različni hipotezi o sliki središča kroga (svetovna ravnina na levi in slikovna ravnina na desni) — $\Delta^{\{P,W\}}\varphi_i = \{P,W\}\varphi_{i+1} - \{P,W\}\varphi_i$.

Tabela 3.1: Primerjava različnih metod ocenjevanja homografije.

<i>Metoda</i>	<i>Zahteve</i>	<i>Prednosti</i>	<i>Pomanjkljivosti</i>
<i>direktna</i>	Ena množica točk na slik s časovnimi oznakami $\{({}^P\mathbf{p}_i, t_i)\}$; znana kotna hitrost ω ali vsaj en cel krog.	Zaključena rešitev; časovno nezahtevno.	Zahteva poznavanje parov točk in kotne hitrosti.
<i>Lourakisova</i>	Vsaj dve množici točk na sliki $\{{}^P\mathbf{p}_i\}$, ki predstavljata dve sliki koplanarnih krogov, ali vsaj dve elipsi ${}^P\mathbf{C}_i$, ki predstavljata sliki koplanarnih krogov.	Ne zahteva poznavanja parov točk; bolj robustno, ker ne temelji na točkah, temveč stožnicah.	Iterativna metoda; računska zahtevnost je pogojena z želeno natančnostjo.
<i>predlagana</i>	Ena množica točk na sliki s časovnimi oznakami $\{({}^P\mathbf{p}_i, t_i)\}$	Zahteva le en sam krog; krog ni nujno poln; ne zahteva poznavanje parov točk; ocenjena je tudi kotna hitrost.	Iterativna metoda; računska zahtevnost je pogojena z želeno natančnostjo.

Lourakisovo metodo, saj se obe metodi poslužujeta podobnega načina iskanja slike središča kroga. Računsko zahtevnost lahko v obeh primerih zmanjšamo, tako da iskalno področje razdelimo najprej z grobo mrežo, nato pa uporabimo bolj fino mrežo le v okolici minimuma. Pri uporabi Lourakisove metode lahko pri tem pride do problema lokalnih minimumov, saj lahko pregroba začetna mreža vrne lokalni namesto globalnega minimuma. Časovna zahtevnost ni posebej kritična, saj ni potrebe za delovanje algoritma v realnem času. Prednost predlagane metode pred Lourakisovo je, da zahteva le en krog. Pokazano je bilo, da lahko Lourakisovo metodo uporabimo ne le na statičnih krogih, vendar tudi na krogih, ki so ustvarjeni dinamično s kroženjem točke. Po drugi strani pa je možno predlagano metodo, ki je bila razvita za uporabo na opazovanju vrteče se točke, uporabiti tudi pri opazovanju določenih statičnih prizorov. Zahteve za uporabo metode v nespremenljivih prizorih so, da imamo na voljo sliko krožnice z nekaj točkami, za katere vemo, kakšni so središčni koti med njimi, vendar pa z direktno meritvijo ne moremo določiti slike središča centra kroga. Lastnosti obravnavanih metod so

zbrane v tabeli 3.1.

3.4.4 Rezultati

Predstavljena metoda ocenjevanja homografije $\mathbf{H}_a \mathbf{H}_p$ je bila preverjena na sintetičnih in realnih podatkih³.

Sintetični podatki

Sintetični podatki za analizo metode so bili pridobljeni s simulacijo gibanja dveh točk, ki sta se vrteli v isti ravnini: ena točka je s kotno hitrostjo $\omega_1 = 0,5 \text{ rad/s}$ opisala krožnico z radijem $r_1 = 0,2 \text{ m}$, druga točka pa je s kotno hitrostjo $\omega_2 = 0,5 \text{ rad/s}$ opisala krožnico z radijem $r_2 = 0,4 \text{ m}$. Meritve so bile pridobljene z vzorčenjem točk na simulirani kameri s časom vzorčenja $T_s = 0,1 \text{ s}$. Preslikavo med ravnino kroženja obeh točk in slikovno ravnino podaja homografija, ki jo želimo oceniti:

$$\mathbf{G} = \begin{bmatrix} 0,4643 & -0,1739 & 0,5317 \\ 0,3255 & 0,4545 & 0,3987 \\ 0 & 0,0003 & 0,0017 \end{bmatrix}. \quad (3.29)$$

Na sintetičnih podatkih je bila opravljena statistična analiza občutljivosti vseh treh opisanih metod na različne vplive. Točnost ocene homografije je bila izmerjena s tremi različnimi merili: merilo točnosti ocene središča vrtenja d (3.30), merilo točnosti prileganja elipse na podatke D (3.31) in merilo odpravljanja perspektivne popačitve z analizo pravokotnosti pravih kotov RAE (3.32). Merili d in D smo uporabili, ker Lourakisova in predlagana metoda temeljita na ocenitvi središča vrtenja in prileganje elipse na podatke. Če je ${}^P\hat{\mathbf{p}}_0$ ocenjen položaj slike središča kroga in je ${}^P\mathbf{p}_0$ prava vrednost slike središča kroga, je merilo d definirano, kot sledi:

$$d = \sqrt{({}^P\hat{\mathbf{p}}_0 - {}^P\mathbf{p}_0)^T ({}^P\hat{\mathbf{p}}_0 - {}^P\mathbf{p}_0)}. \quad (3.30)$$

Merilo točnosti prileganja elipse na sliko kroga je definirano, kot sledi:

$$D = \frac{1}{N} \sum_{j=1}^N \sqrt{({}^P\hat{\mathbf{p}}_j - {}^P\mathbf{p}_j)^T ({}^P\hat{\mathbf{p}}_j - {}^P\mathbf{p}_j)}, \quad (3.31)$$

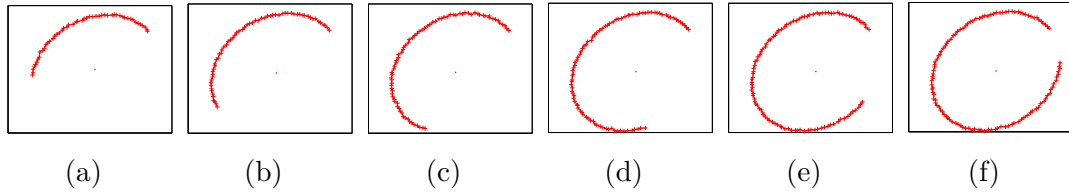
³Predlagana metoda je bila objavljena v članku [143].

kjer leži točka ${}^P\mathbf{p}_j$ na ovojnici prave elipse in točka ${}^P\hat{\mathbf{p}}_j$ na ovojnici ocenjene elipse, pri čemer obema točkama pripada isti kot okoli središča prave elipse. Točke ${}^P\mathbf{p}_j$ in ${}^P\hat{\mathbf{p}}_j$ izberemo tako, da enakomerno razdelimo polni kot okoli središča elipse na N enakih delov (v našem primeru smo izbrali $N = 1000$). Tretje merilo, ki smo ga uporabili, je definirano, kot sledi:

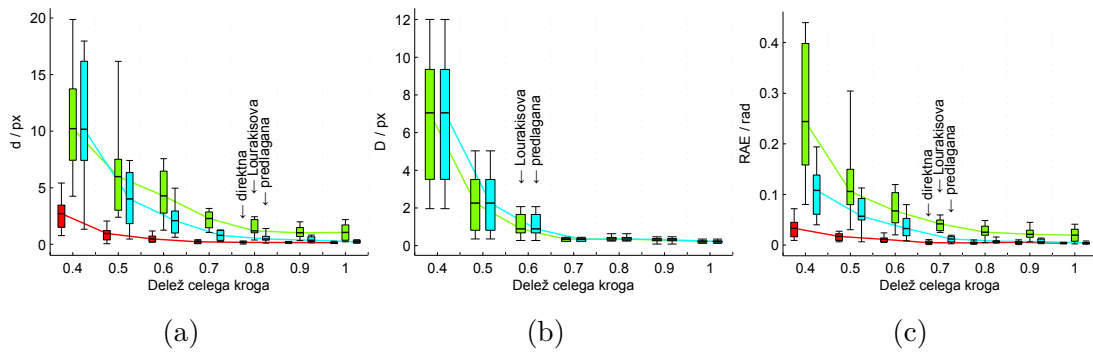
$$RAE = \frac{1}{4} \sum_{j=1}^4 \left| \frac{\pi}{2} - \gamma_j \right|, \quad (3.32)$$

kjer je γ_j kot med premicama, ki ju dobimo s preslikavo slik pravokotnih premic preko ocenjene homografije s slikovne ravnine nazaj v svetovno ravnino. Pri izračunu merila (3.32) upoštevamo štiri premice, ki tvorijo v svetovni ravnini kvadrat in se stikajo v štirih presečiščih, okoli katerih izračunamo štiri kote γ_j , $j \in \{1, 2, 3, 4\}$. Pri popolnoma točno ocenjeni homografiji je vrednost kriterija RAE enaka nič. Od obravnavanih kriterijev je RAE najbolj relevanten, saj neposredno meri točnost odprave perspektivne transformacije. Ostali dve merili (d in D) nam še vedno dajeta dodaten vpogled v točnost kalibracijskega postopka.

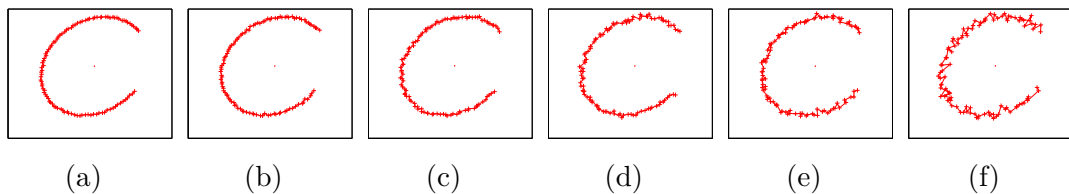
Primerjalna analiza občutljivosti metod pri različnih neidealnih pogojih z merili d , D in RAE je prikazana na slikah 3.6, 3.8, 3.10 in 3.12. Na sliki je 3.6 je predstavljen vpliv dolžine krožnega loka, ki ga opravi točka v ravnini vrtenja glede na en polni krog. Slike uporabljenih podatkov so prikazane na sliki 3.5. Meritve točk na sliki so bile pri tem motene z normalno porazdeljenih belim šumom z varianco $\sigma_n^2 = 1 \text{ px}^2$. Pri uporabi direktne metode smo predpostavili, da je vrednost kotne hitrosti ω znana. Na sliki 3.8 je predstavljen vpliv šuma na meritve položajev točk na sliki pri 0,8 deležu polnega kroga (uporabljeni podatki so zbrani na sliki 3.7). Vpliv šuma na meritev časa vzorčenja je prikazan na sliki 3.10, kjer je uporabljen uniformno porazdeljen šum z maksimalno amplitudo δ_t . V tem primeru je bil znova uporabljen enak delež polne periode (0,8), meritve položajev točk na sliki pa so bile motene z enako varianco šuma kot v primeru spremenljivega deleža kroga. Šum na merjenju časa je prikazan na sliki 3.9. Na sliki 3.12 je prikazan še vpliv naklona slikovne ravnine (kamere) glede na ravnino kroženja. Pri podatkih 3.11 so bili uporabljeni enaki pogoji glede deleža kroga in šuma pri merjenju položajev točk kot v primeru pri spremenljivem šumu pri merjenju časa.



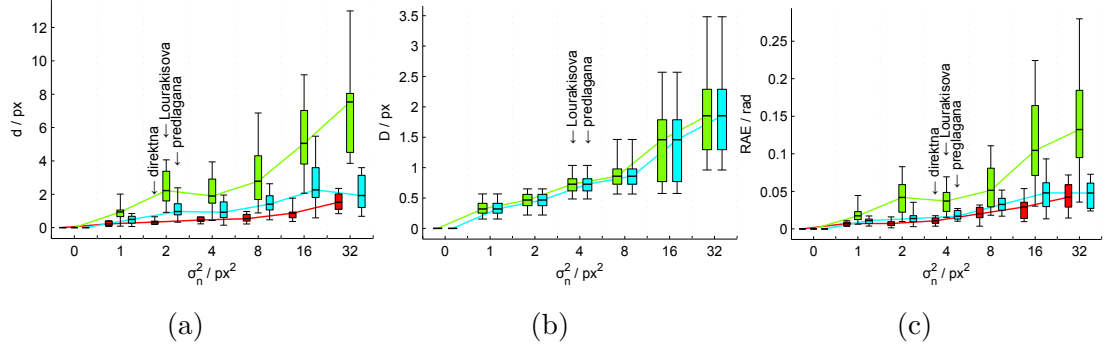
Slika 3.5: Meritve položajev točk na sliki pri različnih deležih obsega celega kroga, ki ga opravi točka v ravnini kroženja, od (a) do (f) so deleži sledeči: 0,4, 0,5, 0,6, 0,7, 0,8 in 0,9.



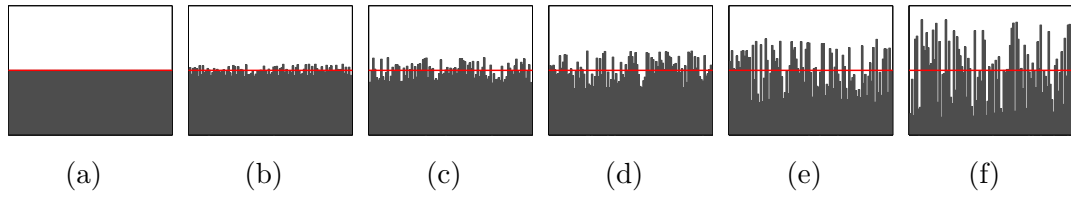
Slika 3.6: Analiza kvalitete ocene pri različnih deležih celega kroga (slika 3.5) za vse tri obravnavane metode z različnimi merili: (a) merilo d , (b) merilo D in (c) merilo RAE .



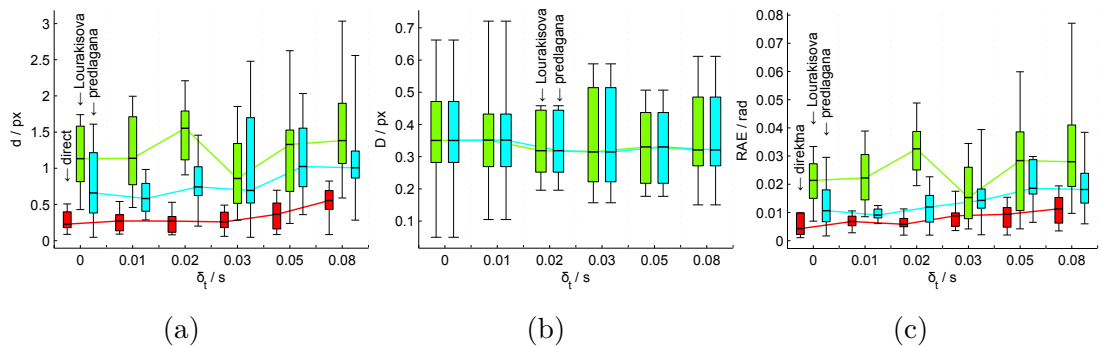
Slika 3.7: Izmerjeni položaji točk na sliki z različnimi variancami normalno porazdeljenega žuma s srednjo vrednostjo nič, od (a) do (f) so variance šuma sledeče: 1 px^2 , 2 px^2 , 4 px^2 , 8 px^2 , 16 px^2 in 32 px^2 .



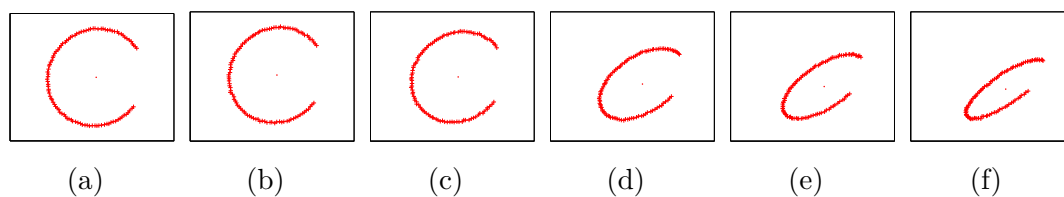
Slika 3.8: Analiza kvalitete ocene pri različnih variancah šuma pri merjenju položajev točk na sliki (slika 3.7) za vse tri obravnavane metode z enakimi merili kot na sliki 3.6.



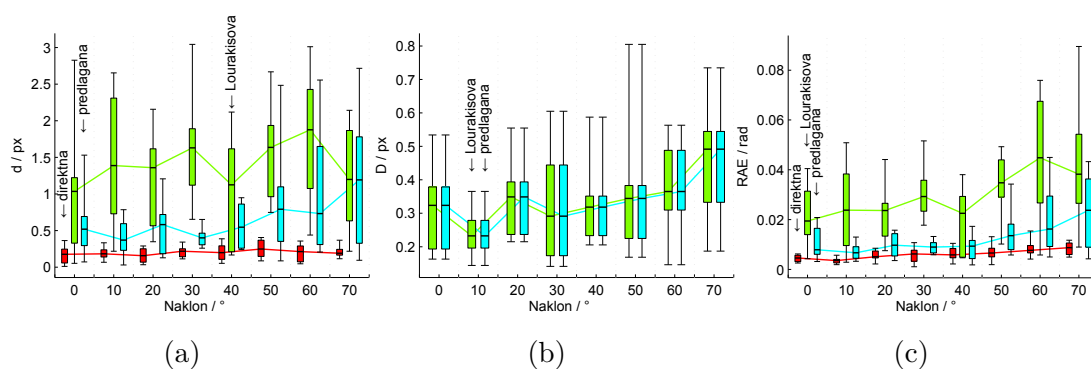
Slika 3.9: Izmerjeni časi z različnimi vrednostmi uniformo porazdeljenega šuma s srednjo vrednostjo nič, od (a) do (f) so vrednosti šuma sledeče: 0 s, 0,01 s, 0,02 s, 0,03 s, 0,05 s in 0,08 s.



Slika 3.10: Analiza kvalitete ocene pri različnih vrednostih šuma pri merjenju časa (slika 3.9) za vse tri obravnavane metode z enakimi merili kot na sliki 3.6.



Slika 3.11: Izmerjeni položaji točk na sliki pri različnih naklonih slikovne ravnine glede na ravnino kroženja, od (a) do (f) so nakloni sledeči: 0° , 10° , 20° , 50° , 60° in 70° .



Slika 3.12: Analiza kvalitete ocene pri različnih naklonih (slika 3.11) za vse tri obravnavane metode z enakimi merili kot na sliki 3.6.

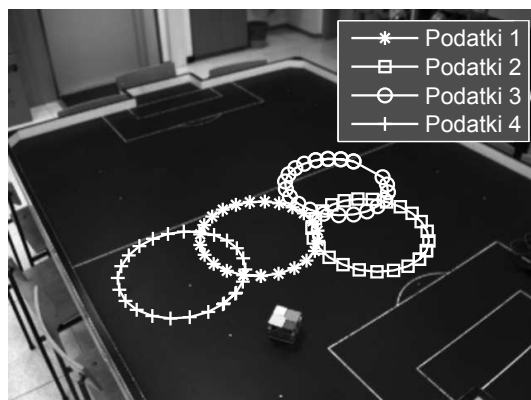
Realni podatki

Metoda je bila primerjalno preizkušena še na realnih podatkih na primeru opazovanja mobilnih sistemov z nekanonično postavitvijo kamere, tako da slikovna ravnina ni vzporedna z ravnino gibanja mobilnih sistemov. Za izvedbo meritev so bili uporabljeni mobilni sistemi, ki se običajno uporabljajo za igranje robotskega nogometa (liga FIRA MiroSot) in barvna kamera z resolucijo 640×480 slikovnih elementov. Na vrhu vsakega mobilnega sistema se je nahajala barvna značka, ki jo je z metodo upravljanja slik v barvnem prostoru (uporaba RGB in HSV barvnih prostorov) možno zaznati in ji slediti na sliki. Posamezni mobilni sistem smo vodili tako, da smo mu poslali konstantni referenčni hitrosti (tangencialna in kotna hitrost sta bili $v_c = 0,4 \text{ m/s}$ in $\omega_c = 2 \text{ rad/s}$). S konstantnim časom vzorčenja ($T_s = 0,1 \text{ s}$) smo nato merili položaj mobilnega sistema na sliki in pri tem zabeležili tudi čas zajema posameznega vzorca. Na sliki 3.13 so predstavljeni zajeti podatki pri štirih različnih postavitvah mobilnega sistema na ravnini. S predstavljenimi postopki smo na podlagi zajetih podatkov ocenili homografijo med ravnino gibanja mobilnega sistema in slikovno ravnino. To nam omogoča odpravo popačitve zaradi perspektivne projekcije oz. izvedbo navideznega pravokotnega pogleda kamere na ravnino⁴. Za izvedbo kalibracije po predlagani in po direktni metodi, smo uporabili le enega izmed izmerjenih krogov, medtem ko smo pri Lourakisovi metodi uporabili preostale kroge za izračun cenilke (3.27). Slika 3.14 prikazuje odstranitev popačitve zaradi perspektivne transformacije z ocenjenimi homografijami. Pri pravilno ocenjeni homografiji bi morale pravokotne črte na igrišču biti pravokotne tudi na sliki, kar nam omogoča, da vizualno ocenimo točnost ocene. V tabeli 3.2 so zbrani še ocenjeni položaji slike središč krogov za vse štiri primere postavitve mobilnega sistema na ravnini.

3.4.5 Diskusija

Na podlagi rezultatov lahko lastnosti predlagane metode v primerjavi z ostalima obravnavanima metodama strnemo v naslednjih ugotovitvah. Predlagana metoda in Lourakisova metoda zahtevata napenjanje elipse na množico točk. Postopek

⁴Odpravo perspektivne popačitve lahko opravimo le za točke, ki se nahajajo v ravnini gibanja barvne značke, ki ji sledimo. Točke, ki ne ležijo v tej ravnini se zato pri izračunu navideznega navpičnega pogleda kamere popačijo.

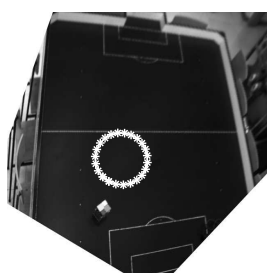


(a)

Slika 3.13: Realni podatki za oceno homografije med ravnino vrtenja (gibanja mobilnega sistema) in slikovno ravnino.

Tabela 3.2: Primerjava ocenjenih slik središča kroga z različnimi metodami na realnih podatkih za štiri različne postavitve mobilnega sistema.

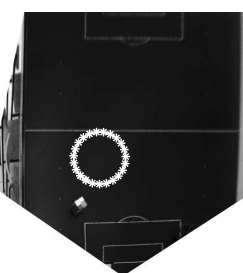
Metoda	Krožnica	Krožnica 1	Krožnica 2	Krožnica 3	Krožnica 4
direktna	$(P_{x_0}, P_{y_0})/px$	(303.522, 276.54)	(441.901, 271.669)	(397.937, 212.223)	(213.39, 318.794)
Lourakisova	$(P_{x_0}, P_{y_0})/px$	(304.512, 274.62)	(438.675, 270.75)	(397.692, 213.389)	(210.997, 317.479)
predlagana	$(P_{x_0}, P_{y_0})/px$	(303.935, 274.511)	(438.954, 270.509)	(397.884, 213.273)	(210.522, 317.299)
	ω/s^{-1}	2.045	2.050	2.046	2.053
direktna*	$(P_{x_0}, P_{y_0})/px$	(303.906, 274.517)	(439.007, 270.512)	(397.919, 213.256)	(210.5, 317.311)



(a)



(b)



(c)



(d)

Slika 3.14: Preslikava ravnine pogleda 3.13 na virtualno ravnino vrtenja, ki je ocenjena z različnimi metodami: (a) z direktno metodo, (b) z Lourakisovo metodo, (c) s predlagano metodo in (d) z direktno metodo ob upoštevanju kotne hitrosti ocenjene s predlagano metodo.

optimalnega prileganja elipse podatkom je občutljiv na šum in dolžino eliptičnega loka; čim krajši je eliptični lok, tem bolj je nenatančen rezultat napenjanja stožnice na podatke — ugotovitev velja splošno, ne glede na uporabljen metodo prileganja stožnice. Kombinacija neprimerno majhnega eliptičnega loka ob določenem šumu lahko privede do popolnoma napačnega prileganja elipse na podatke. Zato je v praksi bolje uporabiti čim daljši eliptični lok. V kolikor je to mogoče, je najbolje uporabiti polni eliptični lok (elipso), kar nam tudi omogoča, da uporabimo direktno metodo za ocenjevanje homografije. Ker v praksi to ni vedno mogoče, je potrebna uporaba predlagane metode. Eksperimentalni rezultati so pokazali, da je tudi ob prisotnosti šuma možno dobiti nepristransko oceno, vendar pa se natančnost ocene z večanjem amplitude šuma poslabšuje. Izkazalo se je, da je vpliv šuma na merjenje časa izrazito manjši kot vpliv šuma na merjenje položajev slik točk. Rezultati so pokazali tudi, da je točnost ocene pri plitvih vpadnih kotih slabša kot pri visokih vpadnih kotih, kar je pričakovano.

4 Strojna obdelava slik

4.1 Nizkonivojska obdelava slik

Slika je dvodimenzionalni signal, kjer vsaka točka v slikovnem prostoru predstavlja bodisi prisotnost oz. odsotnost svetlobe bodisi svetilnost bodisi barvo. Glede na tip signala tako ločimo binarne, sivinske in barvne slike. Na zaslonu kamere nastane realna zvezna slika, ki jo z digitalnim senzorjem pretvorimo v diskretni signal. V računalniškem pomnilniku je slika zapisana kot polje števil, ki so običajno celoštevilskega tipa (običajno nepredznačeno osem- ali šestnajst-bitno celo število, redkeje število s plavajočo vejico). Digitalno sliko lahko matematično predstavimo kot dvodimenzionalno matriko, če gre za binarno oz. sivinsko sliko in kot množico dvodimenzionalnih matrik, če gre za barvno sliko, kjer vsaka matrika predstavlja določen barvni kanal. Barvne slike lahko predstavimo z različnimi barvnimi modeli z različnim številom barvnih kanalov. Pri vodenju na osnovi slike se običajno srečamo z modelom rdeča-zelena-modra (*RGB*), barva-nasičenje-vrednost (*HSV*), barva-nasičenje-svetlost (*HSL*) itd. V nadaljevanju je podrobneje predstavljena le obdelava sivinskih slik. Večino predstavljenih rezultatov lahko uporabimo tudi na barvnih slikah, tako da uporabimo določeno metodo na vsakem barvnem kanalu ločeno.

Analizo slik lahko izvajamo v zveznem ali diskretnem prostoru. Ker je digitalna slika diskreten in kvantiziran signal, se običajno poslužujemo obdelave slike v diskretnem prostoru. V določenih primerih se izkaže, da je bolj primerna obdelava slike v zveznem prostoru [39, 40]. Pri analizi in sintezi metod obdelave slik se velikokrat uporablja zapis slike v zveznem prostoru, medtem ko pri dejanski implementaciji največkrat uporabimo diskretne metode. Obstajajo tudi izjeme, kjer pred začetkom obdelave digitalno sliko opišemo s primerno zvezno funkcijo,

izvedemo obdelavo zveznega signala, po koncu obdelave pa izvedemo vzorčenje rezultata. Tak postopek uporabljajo npr. nekatere metode odvajanja slik [40] in metode za iskanje lokalnih ekstremov [82] itd.

Zvezna slika $W(x, y) \in \mathbb{R}$ je signal, ki je definiran na območju $x \in [0, N]$ in $y \in [0, M]$. Parametra N in M podajata širino in višino slike v zveznem slikovnem prostoru. Koordinatno izhodišče slike je običajno izbrano v levem zgornjem robu slike — os x je usmerjena od leve proti desni, medtem ko je os y usmerjena od zgoraj navzdol. V diskretnem prostoru predstavimo sliko z matriko $\mathbf{V} = [V(i, j)]_{i=1,2,\dots,m;j=1,2,\dots,n}$. Dimenzija matrike $\mathbf{V}$ ustreza velikosti slike, ki jo sestavljajo slikovni elementi v m vrsticah in n stolpcih. Diskretni signal $V(i, j)$ pridobimo z vzorčenjem zvezne slike $W(x, y)$, kar se zgodi pri zajemu slike z digitalnim slikovnim senzorjem. Pri digitalnem zajemu slike se poleg vzorčenja zvezne slike izvede tudi kvantizacija. Vrednosti vzorčenega signala $V(i, j)$ običajno zapišemo s končnim naborom celoštevilskih vrednosti, $V(i, j) \in \mathcal{V} \subset \mathbb{Z}$. Pri obdelavi diskretne slike lahko predpostavimo $V(i, j) \in \mathbb{R}$, torej da vrednosti signala niso kvantizirane. Pri tem se moramo zavedati, da je dejanski signal podvržen šumu kvantizacije. Slika je v pomnilniku računalnika običajno predstavljena z omejenim številom vrednosti (najpogosteje z enim bajtom), zato moramo nameniti posebno pozornost primerni izvedbi metod obdelave digitalne slike na digitalnem računalniku.

Med osnovne metode strojne obdelave slike sodita postopka upravljanja in segmentacije slike, ki omogočata detekcijo področij. O upravljanju govorimo, kadar vrednost slikovnih elementov primerjamo z določeno vrednostjo (pragom). Rezultat upravljanja je običajno binarna slika, kjer je binarna vrednost slikovnih elementov odvisna od izbrane vrednosti praga. S postopkom upravljanja lahko v določenih primerih ločimo določene objekte od ozadja. Kadarkoli imamo opravka s postopkom obdelave slike, ki omogoča določitev slikovnih elementov, ki pripadajo določenemu objektu, govorimo o t. i. segmentaciji. Postopke upravljanja in segmentacije je možno uporabiti na barvnih in sivinskih slikah, rezultat pa je običajno binarna oz. kvantizirana slika. Globalni pristopi upravljanja in segmentacije so običajno primerni le v prilagojenih okoljih, kjer lahko zagotovimo enakomerno osvetljenost [72]. Za primere neenakomerne osvetljenosti so bile razvite adaptivne metode, ki upoštevajo lokalne razmere na sliki oz. v okolju. Eden izmed osnovnih pristopov segmentacije slike, je opis določenega področja s

histogramom, ki nam omogoča segmentacijo slike s postopkom obratne preslikave histograma [39]. Med bolj napredne postopke segmentacije slike sodijo pristopi na osnovi teorije grafov [144–148], ki omogočajo segmentacijo tudi kompleksnih (nehomogenih) objektov.

Slika je dvorazsežni signal, zato lahko nad sliko izvedemo mnoge različne operacije, ki so znane s širšega področja obdelave signalov. Pri obdelavi slik sta zelo pomembni operaciji konvolucije in korelacije. Konvolucija diskretne slike $V(i, j)$ z dvorazsežnim signalom $G(i, j)$ je definirana, kot sledi:

$$U(i, j) = G(i, j) * V(i, j) = \sum_q \sum_p G(p, q) V(i - p, j - q), \quad (4.1)$$

pri čemer je operacija konvolucije v dvorazsežnem prostoru predstavljena z operatorjem $\cdot * \cdot$. Signal $G(i, j)$ predstavlja dvorazsežni filter, ki preoblikuje vhodni signal $V(i, j)$ v izhodni signal $U(i, j)$. Območje seštevanja v (4.1) mora biti dovolj veliko, tako da so upoštevane vse neničelne vrednosti seštevanih vrednosti [39]. Praktično uporabni filtri imajo večino vrednosti, ki doprinesejo h končnemu rezultatu le v omejeni okolici okoli središča filtra. Pri izvedbi konvolucije upoštevamo, da so vrednosti izven te okolice nič, kar omogoča, da pohitrimo izračun konvolucije. Vseeno je izračun konvolucije časovno zahtevna operacija. Znano je, da operaciji konvolucije v slikovnem prostoru ustreza operacija množenja v frekvenčnem prostoru. Z operacijo konvolucije lahko tako izvedemo različne operacije filtriranja slik, kot je npr. glajenje in ostrenje slike. Posebno pomembni so filtri, ki so realni in simetrični. S sodim realnim filtrom je tako možno z operacijo konvolucije izvesti glajenje slike. Z lihim realnim filtrom pa je možno z operacijo konvolucije zaznati robove. Diskretno konvolucijo si lahko predstavljamo tako, da na okolico točke primerno položimo masko filtra in napravimo vsoto zmnožka istoležnih elementov. Pri diskretni konvoluciji, zaradi zahteve po simetričnosti filtra, običajno uporabljamo maske lihih dimenzij. Pomembna je lastnost filtrov, ki omogoča razstavitev filtra na kombinacijo filtrov z manj dimenzijami. Filtre, ki izkazujejo to lastnost, lahko v razstavljeni obliki udejanjimo na računsko mnogo bolj učinkovit način. Konvolucija (4.1) se v primeru dvodimenzionalnega filtra $G(i, j)$, ki ga lahko zapišemo v obliki $G(i, j) = G_1(i)G_2(j)$, poenostavi v [40]:

$$U(i, j) = G(i, j) * V(i, j) = \sum_q G_2(q) \sum_p G_1(p) V(i - p, j - q). \quad (4.2)$$

Operacija podobna konvoluciji je korelacija, ki je definirana, kot sledi:

$$U(i, j) = \sum_p \sum_q G(p, q) V(i + p, j + q). \quad (4.3)$$

Z operacijo korelacije je možno medsebojno primerjati sliki $V(i, j)$ in $G(i, j)$, oz. poiskati položaj določenega vzorca znotraj slike. Operaciji konvolucije in korelacije za izračun vrednosti v določeni točki zahtevata obravnavo signala v okolici te točke, a ker je slika definirana na omejenem območju, je potrebno na robu območja vpeljati dodatne predpostavke. Upoštevamo lahko, da vrednosti zunaj območja slika zavzamejo vrednost na najbližjemu delu roba, da so vrednosti zunaj območja konstantne, ali da je slika periodičen signal [39] — slednje implicitno upoštevamo pri izvedbi Fourierjeve transformacije nad sliko. Lahko pa se preprosto izognemo izračunu konvolucije oz. korelacije na robu območja, če nas vrednosti na robu pretirano ne zanimajo.

Glajenje slik oz. filtriranje slik z nizkopropustnim filtrom je ena izmed bolj pomembnih operacij. Za glajenje slik se največkrat uporablja dvodimenzionalni Gaussov filter:

$$G_\sigma(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right), \quad (4.4)$$

ki ga očitno lahko razstavimo na:

$$G_\sigma(x, y) = \left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right)\right) \left(\frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{y^2}{2\sigma^2}\right)\right). \quad (4.5)$$

Signal $G_\sigma(x, y)$ je zvonaste oblike. Parameter σ^2 predstavlja širino zvona in je na področju statistike bolj znan kot varianca Gaussove funkcije. Diskretno sliko torej zgladimo s konvolucijo z diskretno obliko (4.4), $G_\sigma(i, j) * V(i, j)$. V praksi se pogosto uporabljajo tudi drugi filtri (npr. binomski filter [40]), ki so le približek Gaussovega filtra in omogočajo računsko bolj učinkovito udejanjenje. Zaželeno lastnost filtrov glajenja je, da je določeni integral filtra glajenja čez celotno definicijsko področje enak ena, kar pomeni, da s filtriranjem ne spreminjamo srednje vrednosti signala.

Pri obdelavi slik se pogosto srečamo z izračunom gradienta (4.6), saj lahko na podlagi njegove norme zaznamo robove [39].

$$\nabla W(x, y) = \left[\frac{\partial W(x, y)}{\partial x} \quad \frac{\partial W(x, y)}{\partial y} \right]^T. \quad (4.6)$$

Posamezne približke parcialnih odvodov v (4.6) lahko izračunamo s konvolucijo slike z maskama $\mathbf{G}_A$ in $\mathbf{G}_A^T$ ali z maskama $\mathbf{G}_B$ in $\mathbf{G}_B^T$ [40]:

$$\mathbf{G}_A = \begin{bmatrix} 1 & -1 \end{bmatrix}, \quad \mathbf{G}_B = \begin{bmatrix} 1 & 0 & -1 \end{bmatrix}. \quad (4.7)$$

Robove na sliki lahko nato poiščemo z iskanjem lokalnih maksimumov norme gradienta $\|\nabla V\|$ — običajno uporabimo Evklidsko ali Manhattansko normo. Za iskanje robov lahko uporabimo algoritem angl. „*non-maximum suppression*” [39, 149]. Za detekcijo robov je bilo razvitih mnogo različnih filtrov [39]: diferencialni detektor robov, Prewittov detektor robov, Sobelov detektor robov, Cannyjev detektor robov, Robertov križni detektor robov itd. Detektorje robov se velikokrat uporablja pri segmentaciji področij, saj ravno robovi običajno ločijo različna področja [39].

Robove lahko določimo tudi tako, da poiščemo prehode skozi ničlo drugega odvoda oz. t. i. Laplacea slike. Ker je Laplaceov operator močno občutljiv na šum, sliko pred uporabo Laplaceovega operatorja zgladimo. Običajno uporabimo kar Gaussov filter:

$$\nabla^2 * (G_\sigma * V) = (\nabla^2 * G_\sigma) * V = LoG_\sigma * V. \quad (4.8)$$

Operator LoG_σ je Laplace Gaussove funkcije (angl. „*Laplacian of Gaussian*”) in ima pri obdelavi slik zelo pomembno vlogo. Z normaliziranim operatorjem $\sigma^2 LoG_\sigma$ lahko dosežemo zapis prostora, ki je invarianten na skaliranje [150]. To je zelo pomembna lastnost, saj nam — kot je prikazano tudi v nadaljevanju — omogoča primerjanje vzorcev različnih velikosti. Zelo dober približek $\sigma^2 LoG_\sigma$ je razlika Gaussovih funkcij DoG (angl. „*Difference of Gaussians*”):

$$DoG_{k,\sigma} = G_{k\sigma}(i, j) - G_\sigma(i, j). \quad (4.9)$$

Operator LoG lahko v približku zapišemo kot:

$$\frac{\partial G_\sigma}{\partial \sigma} \approx \frac{G_{k\sigma} - G_\sigma}{k\sigma - \sigma}. \quad (4.10)$$

Predstavljajmo si, da slikovni elementi na sliki podajajo vrednosti temperature v posameznih delih pravokotne površine. Potek spreminjanja temperature s časom lahko opišemo s parcialno diferencialno enačbo za prenos toplote. V kolikor površine ne grejemo z zunanjim virom toplote, postane s časom temperatura po

celotni površini homogena. Časovni potek spreminjanja temperature površine nam na področju obdelave slik predstavlja postopno glajenje slike z dvodimenzionalnim Gaussovimi filtrom. Zato lahko ob upoštevanju parcialne diferencialne enačbe za prenos toplote:

$$\frac{\partial G_\sigma}{\partial \sigma} = \sigma \nabla^2 G_\sigma, \quad (4.11)$$

pridemo do relacije:

$$DoG_{k,\sigma} = G_{k\sigma} - G_\sigma \approx (k - 1)\sigma^2 \nabla^2 G_\sigma. \quad (4.12)$$

Iz (4.12) je razvidno, da operator DoG_σ že izkazuje lastnost normaliziranega operatorja LoG in torej izkazuje invariantnost na skaliranje [82].

4.2 Metode obdelave slik z lokalnimi značilkami

Pri obdelavi signalov se pogosto srečamo z vzorci, ki imajo določen poseben pomen. Opazimo tudi, da se določeni vzorci pogosto ponavljajo. Z obdelavo signalov želimo te vzorce poiskati in jih opisati. Algoritem SIFT (angl. „*Scale Invariant Feature Transform*“) [82] velja za enega izmed prelomnih algoritmov na področju opisa slik z značilkami. Algoritem omogoča opis lokalnih samodejno razpoznanih tipičnih vzorcev z značilkami, ki so invariantne na translacijo, rotacijo, skaliranje, osvetljenost in šum. Celoten algoritem ima značilno tridelno strukturo, ki je skupna mnogim algoritmom za iskanje parov z lokalnimi značilkami. Te faze so: detekcija tipičnih točk oz. vzorcev, opis tipičnih točk z značilkami in iskanje parov med značilkami.

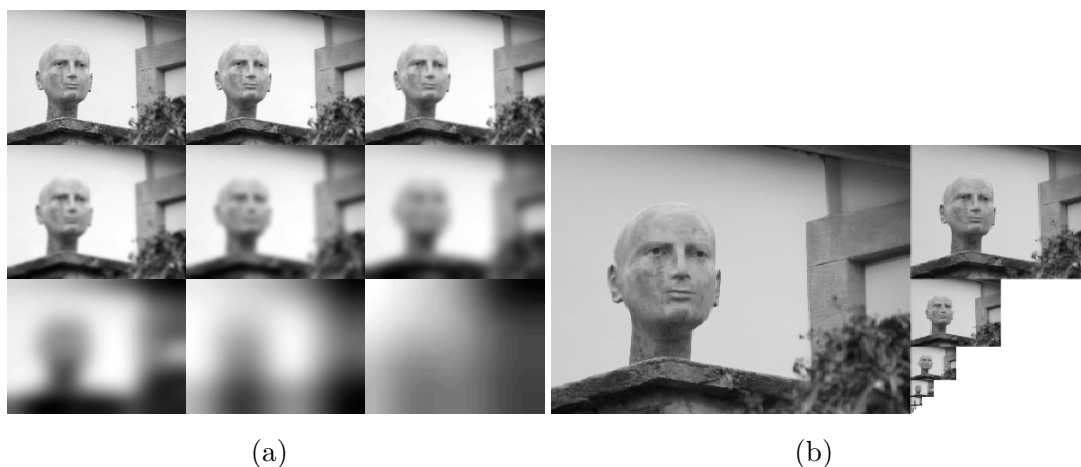
Večino izmed algoritmov, ki so predstavljeni v nadaljevanju, najdemo implementirane v odprtokodni programski knjižnici za strojno obdelavo slik OpenCV [41, 151]. Knjižnica OpenCV združuje večino pomembnejših algoritmov za strojno obdelavo slik in se stalno razvija. Mnogi algoritmi v knjižnici so tudi optimizirani za vzporedno delovanje na več jedrih hkrati, nekatere funkcije pa omogočajo tudi izkoriščanje prednosti posebne strojne opreme, kot so grafične kartice.

4.2.1 Detekcija lokalnih tipičnih vzorcev

V fazi detekcije tipičnih vzorcev želimo na sliki poiskati točke, ki imajo neko določeno tipično okolico, tako da je detekcija takih točk čim bolj ponovljiva in

natančna. Lastnosti ponovljivosti in natančnosti lokalnih tipičnih točk sta dve izmed pomembnih lastnosti detektorja. Detektor, ki izkazuje dobro ponovljivost, zazna veliko število istih tipičnih točk na različnih slikah istega prizora (npr. pri spremenjeni legi pogleda, pri spremenjenih pogojih v osvetlitvi itd.). Detektorji z lastnostjo natančnosti, zaznajo določeno tipično točko vedno na enakem mestu oz. v primerno majhni okolici. Detektorje značilnih točk v grobem ločimo na detektorje robov (npr. FAST [152], AGAST [153]) in detektorje področij (npr. SIFT [82], SURF [83], MSER [85]). Za detektorje robov je značilno, da izkazujejo dobro natančnost, medtem ko je za detektorje področij bolj značilna dobra ponovljivost. Pri detekciji tipičnih vzorcev se pojavi problem določitve velikosti okolice vzorca. Vnaprej ne vemo velikosti vzorca, zato ne vemo, kako veliko okolico moramo izbrati, da bi zaznali določene vzorce. Problem lahko rešimo z uporabo prostora ločljivosti, tako da tipične vzorce iščemo v več različnih ločljivostnih nivojih — tipične vzorce iščemo v t. i. slikovnih piramidah. Določen vzorec, ki se na eni sliki pojavi na določenem mestu v določenem nivoju slikovne piramide, se bo pri premiku in skaliranju tega vzorca, pojavil na nekem drugem mestu na nekem drugem nivoju piramide, ki ustreza opravljenemu premiku in skaliranju. Na tak način je v [82] — kot tudi mnogih drugih algoritmih opisa vzorcev z značilkami — izvedena invariantnost na premik in skaliranje. V [82] so za tipične vzorce izbrane točke, ki jim lahko ponovljivo predpišemo položaj in nivo, obenem pa lahko na podlagi analize okolice ponovljivo določimo tudi njihovo orientacijo. V fazi zaznavanja značilnih točk na sliki poiščemo tipične vzorce, ki jim lahko ponovljivo in natančno priredimo položaj, nivo in orientacijo. Takšna predstavitev omogoča opis najdenih tipičnih točk na način, ki je invarianten na premik, skaliranje in rotacijo. Sedaj pa si podrobneje oglejmo, kako je v algoritmu SIFT dejansko izvedena detekcija tipičnih vzorcev.

Algoritem SIFT detektira tipične vzorce tako, da vhodno sliko najprej prevzorči v vnaprej določeno število oktav (v vsaki naslednji oktavi se razpolovi posamezna dimenzija slike) oz. jo zapiše v obliki Gaussove piramide (slika 4.1b). Nad sliko v vsaki oktavi se nato izvede še konvolucija z spremenljivim Gaussovim filtrom $G_{k,\sigma}$, tako da dosežemo bolj fino razdelitev nivojev Gaussove piramide (slika 4.1a) na s dodatnih nivojev znotraj posamezne oktave (velja $k = 2^{\frac{1}{s}}$). Slika v oktavi o



Slika 4.1: (a) Postopno glajenje slike z Gaussovim filtrom in (b) Gaussova piramida glajenih in decimiranih slik.

v nivoju s je tako:

$$U_{o,s}(i, j) = G_{k\sigma}(i, j) * V_o(i, j), \quad (4.13)$$

kjer je V_o osnovna slika in V pomanjšana slika v oktavo o . Čeprav bi lahko enake rezultate dobili le s konvolucijo slike z Gaussovo funkcijo pri primerno večjem številu nivojev, pa zapis v več oktavah omogoča bolj računsko učinkovito izvedbo algoritma. Tipične točke nato iščemo v prostoru DoG , ki ga dobimo tako, da odštejemo sliki v dveh zaporednih nivojih Gaussove piramide znotraj določene oktave. Tipične točke nato določimo kot ekstreme prostora DoG v lokalni okolici, kjer kot okolico upoštevamo elemente v trenutnem in v dveh sosednjih nivojih Gaussove piramide (okolico sestavlja $9 + 8 + 9 = 26$ slikovnih elementov). Postopek ponovimo za vse nivoje v vseh oktavah. S primerno izbiro števila oktav in nivojev lahko zagotovimo dobro ponovljivost odkritih tipičnih točk [82]. Za čim bolj stabilno lokalizacijo tipičnih točk na odkrite okolice vzorcev napnemo gladko kvadratno krivuljo in položaj posamezne tipične točke določimo kot ekstrem te funkcije (za več informacij glej [82]). Vsaki odkriti tipični točki nato določimo še vsaj eno dominantno orientacijo, ki jo določimo na podlagi histograma gradienta okolice v najbližjem nivoju U . Algoritem SIFT vsebuje tudi postopek za izračun značilnk, ki je opisan v poglavju 4.2.2.

Algoritem za detekcijo in opis slik z značilkami SIFT je dokaj računsko zahteven. Zato je bil razvit algoritem SURF (angl. „*Speeded-Up Robust Features*“) [83], ki

temelji na algoritmu SIFT, le da so bile vpeljane določene dodatne predpostavke in poenostavitve, ki omogočajo hitrejši izračun značilk. Nekateri viri navajajo, da je algoritem SURF po ponovljivosti, stabilnosti in robustnosti primerljiv z algoritmom SIFT, vendar je od njega precej manj računsko zahteven. Vendar je pri tem potrebno upoštevati, da je računska zahtevnost precej odvisna od dejanske implementacije na digitalnem računalniku, tako da obeh algoritmov ni povsem enostavno primerjati.

Algoritem SURF uporablja t. i. integralsko sliko, ki jo določimo iz vhodne slike $V(x, y)$:

$$I(x, y) = \sum_{j=0}^{j \leq x} \sum_{i=0}^{i \leq y} V(i, j). \quad (4.14)$$

S pomočjo integralske slike, lahko vsoto vrednosti slikovnih elementov znotraj pravokotnega okna poljubne velikosti izvedemo le s tremi seštevanji. Ker je zahtevnost postopka neodvisna od velikosti okna, lahko iz integralske slike hitro in enostavno pomanjšamo sliko oz. tvorimo Gaussovo piramido slik; pri tem tudi ne pride do zgibanja frekvenc. Tipične točke določimo z iskanjem maksimuma determinante Hessejeve matrike:

$$H(x, y, \sigma) = \begin{bmatrix} L_{xx}(x, y, \sigma) & L_{xy}(x, y, \sigma) \\ L_{xy}(x, y, \sigma) & L_{yy}(x, y, \sigma) \end{bmatrix}, \quad (4.15)$$

kjer je $L_{xy}(x, y, \sigma)$ konvolucija drugega odvoda Gaussove funkcije $\frac{\partial^2}{\partial x \partial y} G_\sigma(x, y)$ s sliko $V(x, y)$ v točki (x, y) . Ker v praksi uporabimo diskretiziran drugi odvod Gaussove funkcije končnih dimenzij, dobimo pri rotaciji slike za lihe večkratnike števila $\frac{\pi}{4}$ malce slabšo ponovljivost, zato pa je izračun konvolucije slike s filtrom hitrejši. Pri rotaciji slike za večkratnik števila $\frac{\pi}{2}$ dobimo dobro ponovljivost, kar je posledica kvadratne oblike filtra. Izračun drugega odvoda Gaussove funkcije lahko aproksimiramo s Haarovimi funkcijami, s čimer se še nekoliko poenostavi računska zahtevnost algoritma. Detekcije tipičnih vzorcev v različnih ločljivostnih nivojih ne opravimo s postopnim manjšanjem slike, temveč z večanjem filtra; kar je pri uporabi Haarovih funkcij in integralskih slik računsko enostavno in učinkovito. Tipične točke določimo podobno kot pri algoritmu SIFT, le da so posamezni koraki poenostavljeni oz. optimizirani.

Algoritem MSER (angl. „*Maximally Stable Extremal Regions*“) [85] uporablja nekoliko drugačen pristop iskanja značilnih točk. Algoritem stabilno detektira

točke pri spremembi skale, osvetljenosti, rotaciji izven ravnine slike, okluzijah, 3D-translaciji. Zahtevnost algoritma tudi ni prevelika za implementacijo v realnem času. V principu deluje na upravljanju slike s spremenljivim pragom t . Postopek si na sivinski sliki lahko predstavljamo kot film, kjer je časovna spremenljivka prag t . Začetni prizor je bela slika; ko čas teče (prag se spreminja), se na sliki začne pojavljati vedno več črnih področij. Področja, ki so stabilna na določenem intervalu pragov, razglasimo za stabilna področja. Tako smo dobili maksimume (svetlejša področja na temnejšem ozadju); podobno lahko dobimo tudi minimume (temnejša področja na svetlejšem ozadju), če uporabimo inverzno upravljanje.

Za izredno hiter algoritem za detekcijo tipičnih točk se je izkazal algoritem FAST [152]. Algoritem temelji na primerjanju vrednosti slikovnih elementov na krožni okolici okoli centralnega slikovnega elementa (Bresenhamova krožnica s 16 slikovnimi elementi) z vrednostjo praga. Če je določeno minimalno število zaporednih vrednosti slikovnih elementov na krožnici večjih od vrednosti praga, je centralna točka razglašena za tipični vzorec. Algoritem je podoben algoritmu SUSAN [86] za detekcijo vogalov, ki upošteva vse slikovne elemente znotraj določene okolice. Preverjanje vrednosti v okolici centralnega slikovnega elementa ni izvedeno zaporedno po elementih, temveč na optimalen način, tako da je število preverjanj čim manjše. Izboljšavo algoritma FAST predstavlja algoritem AGAST [153], ki naj bi bil bolj primeren za splošno vsebino na sliki.

Glede na lastnosti in način delovanja so posamezni algoritmi detekcije tipičnih vzorcev bolj ali manj primerni za različne slikovne vsebine. Izbira detektorja tipičnih vzorcev je tako odvisna od namena uporabe. Eksperimentalno primerjavo med nekaterimi detektorji lahko najdemo npr. v [84].

4.2.2 Opis z značilkami

V prejšnjem poglavju so predstavljene metode za odkrivanje tipičnih vzorcev na sliki, v tem poglavju je prikazano, kako te vzorce oz. okolice tipičnih točk opišemo. Vzorce predstavimo z značilkami, tj. z vektorji značilk. Vzorce želimo opisati na čim bolj diskriminatorenen način. To pomeni, da želimo, da je opis podobnih vzorcev podoben in čim bolj različen od preostalih vzorcev. Obenem želimo, da je predstavitev vzorca robustna na določen nabor geometrijskih transformacij (translacijo, rotacijo, skaliranje, strig, perspektivno transformacijo itd.) in

na določene spremenljive pogoje, ki so posledica spremenljive osvetlitve, šuma, neostrości, itd. Invariantnost značilk na translacijo, rotacijo in skaliranje sodi med pomembnejše, saj s temi preslikavami lahko opišemo poljubno linearno (afino) dvodimenzionalno preslikavo. Izkaže se, da je običajno možno uporabiti takšne značilke tudi pri ostalih nelinearnih preslikavah, saj lahko preslikavo med lokalnimi okolici točk mnogokrat obravnavamo kot linearno. Z detektorjem značilnih točk smo torej določili položaje, velikosti in orientacije tipičnih vzorcev. Okolico vsakega zaznanega tipičnega vzorca lahko tako opišemo v njegovem lokalnem koordinatnem sistemu, ki ima velikost enote sorazmerno z velikostjo vzorca.

V algoritmu SIFT [82] je na vsako okolico položena kvadratna mreža (velikost mreže je odvisna od velikosti vzorca, orientacija od dominantne smeri vzorca), ki je razdeljena na $4 \times 4 = 16$ delov. Vsak del je nato razdeljen še na več točk v katerih se izračuna smer in absolutna vrednost gradienta slike. Na podlagi izračunanih gradientov v vseh točkah znotraj enega dela se določi histogram orientacij gradienta z osmimi celicami. Dobljeni histogram sestavlja eno šestnajstino vektorja značilk, ostale elemente vektorja značilk dobimo tako, da izračunamo histogram še v preostalih 15 delih na vzorec položene kvadratne mreže. Predstavitev s histogrami je v strojnem vidu pogosta, histogram orientacij gradienta srečamo npr. v algoritmu za detekcijo ljudi na slikah [154]. Vrednosti histogramov so še normirane kar zagotovi invariantnost značilk na svetlost. Vektor značilk tako sestavlja 128 vrednosti. Za primerjanje značilk se uporablja Evklidska norma.

Algoritmu SIFT je podoben tudi algoritem GLOH, ki daje nekoliko boljše rezultate, a je računsko nekoliko bolj zahteven. Na podoben način kot pri algoritmu SIFT se določijo tudi značilke v algoritmu SURF [83], ki je znan kot optimizirana in poenostavljena verzija algoritma SIFT. Zopet na izbrano okolico položimo mrežo velikosti 4×4 . Na več mestih znotraj posameznega dela mreže izračunamo odziv slike na Haarovo funkcijo prvega reda v dveh ortogonalnih smereh. Vrednosti vektorja značilk dobimo tako, da napravimo vsoto vseh odzivov in absolutnih vrednosti odzivov znotraj enega dela po obeh smereh. Tako za vsak del dobimo štiri vrednosti, ki jih vpišemo v vektor značilk, kar da vektor s 64 vrednostmi. Za primerjanje značilk zopet uporabimo Evklidsko normo.

Zapis značilk z realnimi števili (s plavajočo ali fiksno vejico) je precej prostorsko potraten. Obenem je primerjava značilk z Evklidsko normo dokaj časovno zahtevna operacija. Zato so bile razvite metode za zmanjšanje dimenzij vektorjev značilk,

kot je npr. PCA-SIFT [155]. Algoritem PCA-SIFT s pomočjo metode glavnih komponent doseže primerljive rezultate z osnovnim algoritmom SIFT, čeprav uporabi le značilke s 36 elementi. Slabost pristopa je, da je izračun optimalnega nabora značilk računsko zahteven, saj je analizo značilk potrebno izvesti na osnovnem naboru značilk. Obstajajo še drugi algoritmi za zapis značilk na bolj prostorsko in računsko učinkovit način, kot je npr. pretvorba značilk v binarne nize [39]. Večina takšnih načinov zahteva obdelavo osnovnih značilk, kar zahteva predhodni izračun osnovnih visokodimenzijskih značilk, da določimo nizkodimenzijske značilke. A vsi pristopi niso takšni. Obstajajo algoritmi, ki omogočajo izračun t. i. binarnih značilk neposredno na podlagi lokalne okolice vzorca. Binarne značilke zahtevajo precej manj pomnilniškega prostora za zapis kot običajne (realne) značilke. Določene binarne značilke omogočajo tudi precej učinkovito primerjanje podobnosti, saj lahko kot mero uporabimo kar Hammingovo razdaljo. Torej moramo medsebojno primerjati le bite v značilkah in prešteti število razlik. Takšne operacije so še posebej primerne za izvedbo na digitalnem računalniku, obenem pa imajo sodobni procesorji vgrajene posebne ukaze za izredno učinkovito izvedbo takšnih operacij.

V to kategorijo spadajo značilke BRIEF (angl. „*Binary Robust Independent Elementary Features*”) [87]. Vektor značilk določimo tako, da okolico najprej zgladimo z Gaussovim filtrom. Nato izberemo množico parov točk v okolici in izvedemo primerjavo vrednosti svetlosti v vsakem paru, rezultat primerjanja pa shranimo v vektor značilk. Položaje točk v parih izberemo na nek naključen način. V [87] je predlagano nekaj različnih pristopov, ki jih lahko uberemo pri izbiri položajev točk v parih — npr. porazdelitev položajev točk z Gaussovo porazdelitvijo okoli središča vzorca. Običajno je dolžina vektorja značilk 128, 256 ali 512, kar ustreza tudi številu bitov potrebnih za zapis vektorja značilk. Značilke BRIEF so enostavno izračunljive in omogočajo enostavno primerjanje s Hammingovo razdaljo.

Značilke BRIEF niso invariantne na rotacijo. Na osnovi značilk BRIEF in detektorja FAST [75], so bile izdelane značilke ORB (angl. „*Oriented FAST and Rotated BRIEF*”) [88], ki izkazujejo invariantnost na rotacijo. Metoda dejansko odpravi pomanjkljivost detektorja FAST, ki zaznanim tipičnim vzorcem ne predpiše dominantne smeri. Algoritem vsakemu tipičnemu vzorcu dodeli dominantno smer na podlagi izračuna slikovnih momentov prvega reda, kar omogoča da izračunamo

značilke BRIEF na način, ki je invarianten na rotacijo.

V [156] so predstavljene binarne značilke BRISK (angl. „*Binary Robust Invariant Scalable Keypoints*”), ki so invariantne tako na rotacijo kot skaliranje. Kot smo že omenili vsebuje algoritem BRISK tudi algoritem za izračun tipičnih vzorcev. Vektor značilk določimo podobno kot v primeru BRIEF, le da so položaji točk razporejeni v vzorec pravilne oblike, ki je orientiran glede na dominantno smer tipičnega vzorca. Vektor značilk je dolg 512 bitov.

Razvoj nekaterih značilk so navdahnili biološki sistemi. Med slednje sodijo binarne značilke FREAK (angl. „*Fast Retina Keypoint*”) [89]. Značilke posnemajo delovanje mrežnice v očesu. Te značilke so definirane podobno kot značilke BRISK, saj so tudi v tem primeru vrednosti vektorja značilk določene v točkah, ki tvorijo vzorec pravilnih oblik. Gostota razporeditve točk v vzorcu naj bi posnemala razporeditev fotoreceptorjev na mrežnici v očesu.

4.2.3 Iskanje parov

Iskanje parov točk med dvema slikama je eden izmed temeljnih problemov v strojnem vidu. Veliko aplikacij v strojnem vidu temelji na iskanju parov točk med slikami. Na podlagi parov točk lahko ocenimo globino prizora, rekonstruiramo 3D-obliko predmetov in 3D-prostor, izvedemo detekcijo in razvrščanje objektov, ocenimo relativni premik in določimo lego kamere v prostoru, sledimo objektom, itd. Reševanje iskanja parov točk med slikami je izredno zahtevna naloga, kar je posledica perspektivne preslikave, distorzij leč, digitalne narave senzorja, različnih oblik šuma, spremenljivih pogojev v okolici, okluzij, zamegljenosti, ponavljajočih se vzorcev, homogenih področij itd.

Pristope za iskanje parov lahko razdelimo v skupino za gosto iskanje parov in skupino za redko iskanje parov. Pri gostem iskanju parov želimo vsaki točki na sliki določiti njen par na drugi sliki, če je le viden. Pri redkem iskanju parov, je cilj določiti le pare med redkima množicama točk na obeh slikah. Ti dve množici pridobimo s postopkom detektiranja tipičnih vzorcev. Pare med tipičnimi vzorci tako poiščemo s primerjavo značilk, pri čemer moramo uporabiti ustrezno mero.

Algoritem SIFT uporablja za mero podobnosti med značilkami Evklidsko normo. Za par izberemo tisti dve značilki, ki sta si najbolj podobni oz. ima razlika njunih vrednosti najmanjšo normo med vsemi značilkami. Pri iskanju parov se ne

moremo izogniti napačnemu razvrščanju parov. Da se temu čim bolj izognemo, običajno pri iskanju zahtevamo, da drugi najboljši par ni preblizu najboljšemu paru. Pogosto se uporablja tudi dvosmerno iskanje, kjer vsakemu elementu iz prve množice poiščemo najboljši par v drugi množici in par sprejmemo kot pravilen le, če najdemo isti par pri enakem iskalnem postopku, le da iskanje obrnemo in vsakemu elementu iz druge množice poiščemo par v prvi množici. Obstajajo še mnoge druge oblike preverjanja parov [40]. V nadaljevanju so prikazane še metode za robustno iskanje parov, ki temeljijo na uporabi modela.

4.3 Sledenje vzorcev skozi sekvenco slik

Kadar opazujemo smeri in magnitude premikov skozi mirujočo množico točk na sliki, govorimo o t. i. *optičnem toku* (angl. „*optical flow*”) [18, 157, 158]. Optični tok mnogokrat izračunamo na enakomerno razporejeni množici točk, zato gre v tem primeru za gosto sledenje vzorcev. V kolikor sledimo premikom množice točk oz. vzorcev skozi sekvenco slik (zanimajo nas trajektorije vzorcev), gre za t. i. *sledenje vzorcev* (angl. „*feature tracking*”) [37]. Sledenje vzorcev je običajno izvedeno le na dokaj redki množici točk znotraj slike (npr. na tipičnih vzorcih). Za sledenje vzorcev obstajata dve metodi: *neodvisna* in *odvisna* [159]. Pri sledenju vzorcev z neodvisno metodo poiščemo na obeh slikah tipične vzorce neodvisno, nato pa poiščemo pare med vzorci, s čimer ugotovimo premike posameznih vzorcev. Pri sledenju z odvisno metodo vsakemu vzorcu na prvi sliki poiščemo par na drugi sliki bodisi v lokalni okolici trenutnega vzorca bodisi v napovedani okolici, pri čemer premik poiščemo z določeno optimizacijsko metodo. Relativni premik okolice lahko napovemo na podlagi modela gibanja, primerjavo med vzorci pa napravimo npr. s korelacijo, za optimizacijo pa uporabimo metodo najstrmejšega sestopa (angl. „*steepest descent method*”). Sledenje z neodvisno metodo je bolj primerno za velike premike, medtem ko je sledenje z odvisno metodo primerno za majhne premike. V kolikor imamo na voljo dovolj točen model gibanja, lahko uporabimo odvisno metodo tudi za velike premike, čeprav se pri prevelikih premikih zaradi omejene natančnosti modela pojavi problem lokalnih minimumov. Primer algoritma za sledenje vzorcev z odvisno metodo je algoritem Kanade-Lucas-Tomasi (KLT) [81]. V nadaljevanju tega poglavja je podrobneje predstavljeno sledenje vzorcev, ki se v

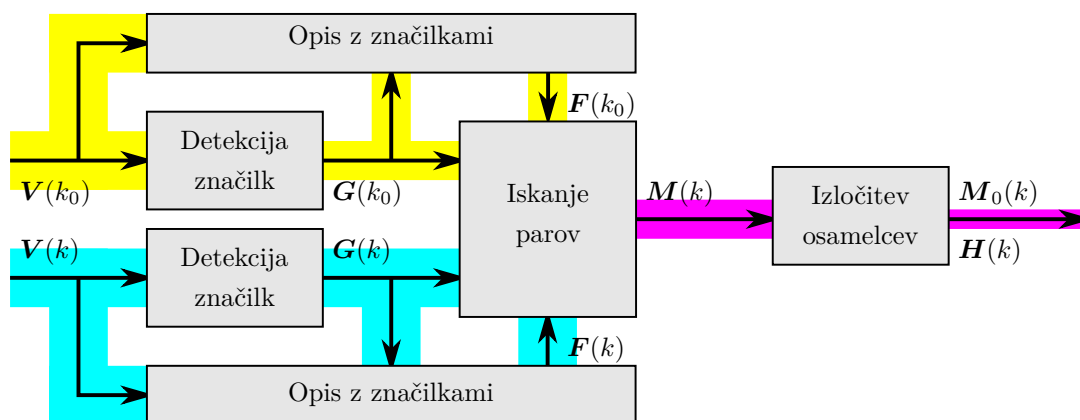
trirazsežnem prostoru nahajajo na ravnini, z neodvisno metodo.

Predpostavimo, da s kamero opazujemo ravnino z dveh različnih leg, tako da je določen del ravnine viden na obeh slikah. Na podlagi meritev vzorcev na obeh slikah želimo poiskati preslikavo, ki povezuje obe sliki. Predpostavimo, da imamo opravka z eno kamero, ki je časovno porazdeljena, s katero zajamemo dve sliki — prvo v trenutku k_0 in drugo v trenutku $k > k_0$. Iščemo povezavo med točkama ${}^P\mathbf{p}(k_0) \in \mathbb{R}^2$ in ${}^P\mathbf{p}(k) \in \mathbb{R}^2$, ki pripadata isti točki v svetovnem koordinatnem sistemu ${}^W\mathbf{p} \in \mathbb{R}^3$. Kot že vemo, lahko preslikavo med točkama opišemo s homografijo (3.4):

$${}^P\mathbf{p}(k) \simeq \mathbf{H}(k, k_0) {}^P\mathbf{p}(k_0). \quad (4.16)$$

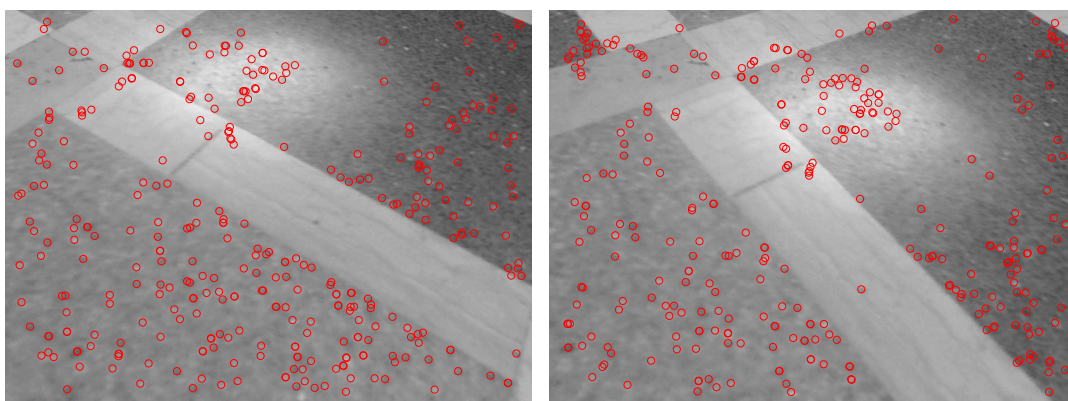
Homografijo med ravninama lahko ocenimo na podlagi znanih parov točk, ki pripadata obema ravninama (poglavje 3.3). Za oceno preslikave so potrebni najmanj štirje pari točk. Običajno uporabimo večje število parov in napravimo oceno v smislu najmanjših kvadratov. Vendar s številom parov ni niti potrebno niti smiselno pretiravati, saj običajno dobimo zadovoljive rezultate že pri relativno redki množici primerno natančnih parov; večje število parov pa le večja računsko zahtevnost ocenjevanja modela. Torej ugotovimo, da ni potrebno za vsako točko na ravnini (sliki) poiskati njenega para na drugi ravnini (sliki), in se tako lahko poslužujemo le redke množice parov. Ker nas ne zanima gosta množica parov se primerno zmanjša računska zahtevnost, saj moramo poiskati le primerno majhno redko množico parov. Pri iskanju parov se nam pogosto zgodi, da določene pare napačno razvrstimo. Ker le-ti lahko močno vplivajo na točnost ocene, se moramo takšnih parov znebiti. V nadaljevanju je prikazano, kako lahko s pomočjo modela robustno ocenimo parametre modela, tudi ko imamo veliko število napačnih parov. Glede računske zahtevnosti odstranjevanja napačno razvrščenih parov pa nam zopet pride prav, da imamo opravka le z redko množico parov.

V poglavju 4.2 smo si ogledali metode, ki jih lahko uporabimo za iskanje parov med tipičnimi točkami, ki jih opišemo z značilkami. Shema iskanja korespondenčnih parov med dvema slika je prikazana na sliki 4.2, ki vsebuje bloke za detekcijo tipičnih vzorcev (poglavje 4.2.1), opis tipičnih vzorcev z značilkami (poglavje 4.2.2) in iskanje parov med značilkami (poglavje 4.2.3). Na sliki 4.3 so prikazane detektirane značilke, ki jih vrne algoritem SIFT. S primerjavo značilk lahko poiščemo pare med tipičnimi točkami, kar je prikazano na slik 4.4a. Opazimo, da je



Slika 4.2: Shema za sledenje tipičnih vzorcev skozi sekvenco slik.

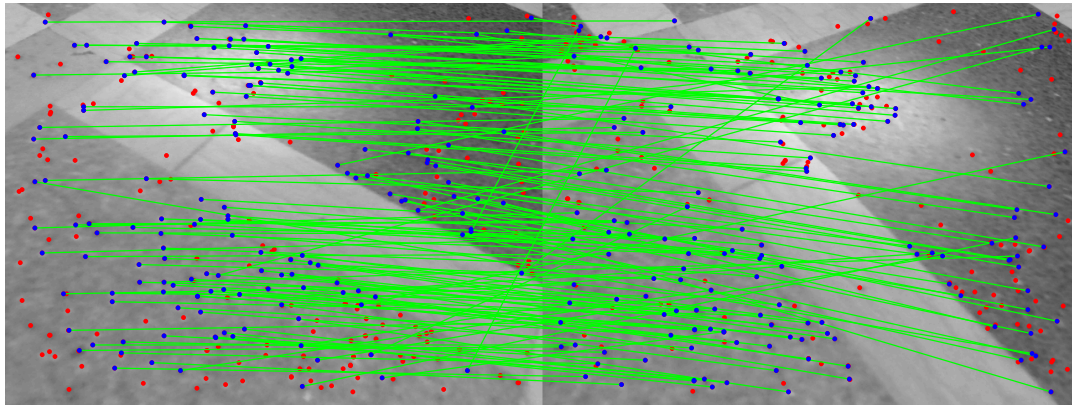
med pravilno najdenimi pari kar nekaj napačno razvrščenih parov, t. i. *osamelcev*. Da omejimo število osamelcev, lahko pri iskanju parov uvedemo dodatne ukrepe, kot je dvosmerno iskanje parov, preverjanje najboljših nekaj parov itd. Napačno razvrščene pare lahko na učinkovit način izločimo z algoritmom RANSAC [94] za robustno ocenjevanje modela, ki omogoča pravilno oceno modela tudi, ko imamo opravka z velikim številom osamelcev (tudi več kot 50 odstotkov). Za uporabo algoritma RANSAC moramo poznati strukturo modela, ki ga želimo oceniti. Ker smo predpostavili, da opazujemo ravnino, imamo opravka s preslikavo med ravninama, ki jo lahko matematično opišemo s homografijo (4.16). S slike 4.4b je razvidno, da algoritem RANSAC z modelom homografije omogoča izločitev osamelcev. S tako ocenjenim modelom lahko izvedemo natančno poravnavo slik (slika 4.5), kar kaže na učinkovitost algoritma. Zaradi svoje učinkovitosti, je uporaba robustnega ocenjevanja modela z algoritmom RANSAC oz. eno izmed njegovih različic na področju strojnega vida zelo pogosta.



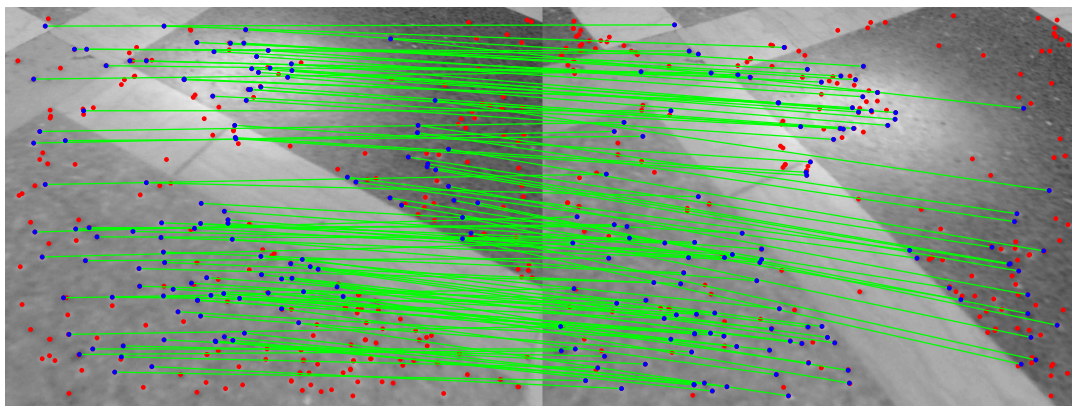
(a)

(b)

Slika 4.3: Detektirane tipične točke z algoritmom SIFT pri dveh premaknjenih postavitvah mobilnega sistema s kamero.

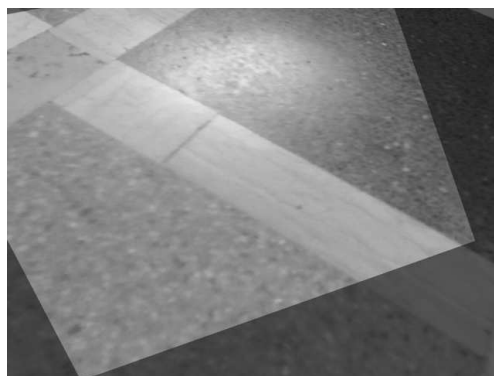


(a)



(b)

Slika 4.4: (a) Najdeni pari med značilkami na slikah 4.3a in 4.3b z metodo primerjanja vektorjev značilk in (b) robustno ocenjeni pari točk z algoritmom RANSAC.



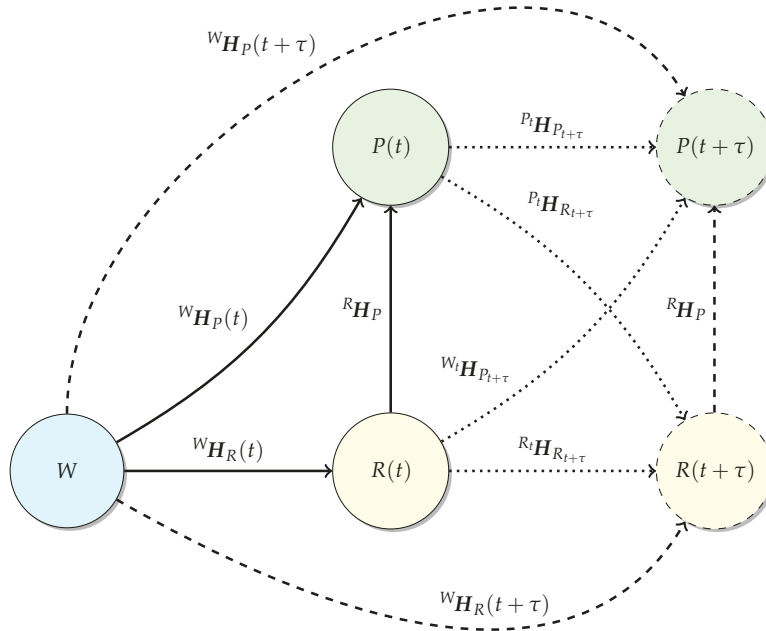
Slika 4.5: Preslikava slike 4.3b na sliko 4.3a.

4.4 Ocena premika iz sledenja značilk

V poglavju 4.3 je prikazano, kako lahko pri opazovanju ravnine iz različnih pogledov na robusten način najdemo pare točk med slikami. Na podlagi parov točk lahko nato tudi ocenimo preslikavo oz. homografijo med slikama. V poglavju 4.3 je bilo predpostavljeno le, da imamo dve časovno porazdeljeni kameri, ki opazujeta ravnino iz dveh različnih leg v prostoru. V tem poglavju je opisana situacija, ko je kamera nameščena na mobilni sistem, ki se giblje po ravni površini, ki jo opazujemo s kamero. Predpostavimo, da je postavitev kamere glede na mobilni sistem poljubna in nespremenljiva. Pri premiku mobilnega sistema v ravnini se spremeni tudi pogled kamere. Na sliki 4.3 sta dejansko prikazani dve sliki ravnine, ki smo jo zajeli na tak način. V tem poglavju so podrobneje predstavljene preslikave, ki povezujejo opise značilk v različnih koordinatnih sistemih.

Na sliki 4.6 so predstavljene povezave med svetovnim koordinatnim sistemom W , koordinatnim sistemom mobilnega sistema R in koordinatnim sistemom slike P v časovnih trenutkih t in $t + \tau$. Na sliki je zaradi preglednosti izpuščen koordinatni sistem kamere C , ki si ga lahko predstavljamo na mestu koordinatnega sistema P , saj predpostavljamo, da med njim velja statična transformacija (nespremenljivi notranji parametri kamere). Predstavimo sedaj matematični opis preslikav, ki nastopajo na sliki 4.6. Zanimajo nas preslikave med naslednjimi koordinatnimi sistemi: med svetovnim koordinatnim sistemom W , lokalnim koordinatnim sistemom mobilnega sistema R in koordinatnim sistemom slike P ; ter med časovno zamaknjenim koordinatnim sistemom slike in koordinatnim sistemom mobilnega sistema. V vseh primerih obravnavamo le preslikave med ravninami, saj smo predpostavili, da se mobilni sistem giblje po ravnini, ki jo opazujemo s kamero, na kateri nastane slika na ravnem zaslonu. Splošno perspektivično preslikavo med ravninama lahko matematično opišemo s homografijo. Pri obravnavi smo se omejili le na preslikavo točk, saj s tem ne izgubimo na splošnosti, ker — kot je predstavljeno v poglavju 3.1 — so na podlagi znane preslikave med točkami znane tudi preslikave med ostalimi stožnicami.

Predpostavimo, da je preslikava med koordinatnim sistemom slike in lokalnim koordinatnim sistemom mobilnega sistema podana s časovno nespremenljivo homografijo ${}^R\mathbf{H}_P$ — preslikava je izražena glede na koordinatni sistem mobilnega sistema. V kolikor podaja preslikava ${}^W\mathbf{H}_R(t)$ lego mobilnega sistema v svetov-



Slika 4.6: Koordinatni sistemi (vozlišča) in preslikave med njimi (povezave).

nem koordinatnem sistemu ob času t , lahko izrazimo določeno točko v različnih koordinatnih sistemih. Čeprav lahko položaj poljubne točke $\mathbf{p}$ opišemo v različnih koordinatnih sistemih, pri čemer velja splošna zveza:

$${}^W\mathbf{p}(t) \simeq {}^W\mathbf{H}_R(t) {}^R\mathbf{p}(t) \simeq {}^W\mathbf{H}_R(t) {}^R\mathbf{H}_P {}^P\mathbf{p}(t) \simeq {}^W\mathbf{H}_P(t) {}^P\mathbf{p}(t), \quad (4.17)$$

lahko ločimo tudi nekaj posebnih primerov, ki so praktično uporabni.

Predpostavimo, da položaj točke $\mathbf{p}$ glede na lokalni koordinatni sistem mobilnega sistema miruje, torej ${}^R\mathbf{p}(t + \tau) = {}^R\mathbf{p}(t) = {}^R\mathbf{p}$ za vsak čas $\tau \in \mathbb{R}$. Glede na predpostavke je točka $\mathbf{p}$ tudi v koordinatnem sistemu slike časovno nespremenljiva, torej ${}^P\mathbf{p} \simeq {}^P\mathbf{H}_R {}^R\mathbf{p}$. S stališča svetovnega koordinatnega sistema pa se točka $\mathbf{P}$ premika, saj velja:

$${}^W\mathbf{p}(t) \simeq {}^W\mathbf{H}_R(t) {}^R\mathbf{p} \simeq {}^W\mathbf{H}_R(t) {}^R\mathbf{H}_P {}^P\mathbf{p} \simeq {}^W\mathbf{H}_P(t) {}^P\mathbf{p}. \quad (4.18)$$

Vpeljemo lahko tudi nekoliko drugačno predpostavko. Recimo, da točka $\mathbf{p}$ miruje glede na svetovni koordinatni sistem. Torej je v tem primeru ${}^W\mathbf{p}(t + \tau) = {}^W\mathbf{p}(t) = {}^W\mathbf{p}$ za vsak $\tau \in \mathbb{R}$. S stališča mobilnega sistema, se točka $\mathbf{P}$ premika

tako v koordinatnem sistemu mobilnega sistema kot tudi v koordinatnem sistemu slike. Velja:

$${}^W\mathbf{p} \simeq {}^W\mathbf{H}_R(t) {}^R\mathbf{p}(t) \simeq {}^W\mathbf{H}_R(t) {}^R\mathbf{H}_P {}^P\mathbf{p}(t) \simeq {}^W\mathbf{H}_P(t) {}^P\mathbf{p}(t). \quad (4.19)$$

Kadar imamo opravka s togimi telesi, vedno izberemo vsaj en koordinatni sistem na tem telesu, tako da točke togega telesa v takšnem koordinatnem sistemu mirujejo. Tako nas v splošnem zanimajo le preslikave med koordinatnimi sistemi, ki omogočajo preslikavo ne le točk, temveč tudi ostalih stožnic (glej poglavje 3.1). Iz primerov (4.17), (4.18) in (4.19) lahko izluščimo splošno zvezo:

$${}^W\mathbf{H}_P(t) \simeq {}^W\mathbf{H}_R(t) {}^R\mathbf{H}_P, \quad (4.20)$$

ki podaja povezavo med obravnavanimi koordinatnimi sistemi v določenem časovnem trenutku t . Na sliki 4.6 so tako grafično prikazane podane relacije za časovni trenutek t (povezave s polno črto) in še za časovni trenutek $t + \tau$ (povezave s črtkano črto). S pikčasto črto so podane relacije med koordinatnimi sistemi v dveh različnih časovnih trenutkih — v trenutkih t in $t + \tau$ —, ki so predstavljene v nadaljevanju.

Preslikava ${}^{R_t}\mathbf{H}_{R_{t+\tau}}$ podaja lego lokalnega koordinatnega sistema mobilnega sistema ob času $t + \tau$ glede na isti koordinatni sistem ob času t . Podobno predstavlja preslikava ${}^{P_t}\mathbf{H}_{P_{t+\tau}}$ povezavo med časovno premaknjenim koordinatnim sistemom slike. S pomočjo slike 4.6 lahko enostavno pridemo do relacije:

$${}^{R_t}\mathbf{H}_{R_{t+\tau}} \simeq {}^R\mathbf{H}_P {}^{P_t}\mathbf{H}_{P_{t+\tau}} {}^R\mathbf{H}_P^{-1}. \quad (4.21)$$

Torej je preslikava točke ${}^P\mathbf{p}(t)$ v časovni trenutek $t + \tau$ podana s pravilom:

$${}^P\mathbf{p}(t + \tau) \simeq {}^{P_t}\mathbf{H}_{P_{t+\tau}}^{-1} {}^P\mathbf{p}(t) \simeq {}^{P_t}\mathbf{H}_{P_{t+\tau}}^{-1} {}^R\mathbf{H}_P^{-1} {}^W\mathbf{H}_R^{-1}(t) {}^W\mathbf{p}(t). \quad (4.22)$$

Enačba (4.21) podaja povezavo med premikom na sliki in premikom mobilnega sistema — s pomočjo te enačbe lahko odpravimo popačitev zaradi perspektivne projekcije. Lega mobilnega sistema v svetovnem koordinatnem sistemu ob času $t + \tau$ je torej podana s homografijo ${}^W\mathbf{H}_R(t) {}^{R_t}\mathbf{H}_{R_{t+\tau}}$, kot produkt homografije začetne lege (ob času t) in relativnega premika, ki nastane v času $\tau > 0$. Za matriki ${}^W\mathbf{H}_R(t)$ in ${}^{R_t}\mathbf{H}_{R_{t+\tau}}$ predpostavimo, da imata posebno obliko:

$$\mathbf{H}(x(t), y(t), \varphi(t), s(t)) = \begin{bmatrix} s(t) \cos \varphi(t) & -s(t) \sin \varphi(t) & x(t) \\ s(t) \sin \varphi(t) & s(t) \cos \varphi(t) & y(t) \\ 0 & 0 & 1 \end{bmatrix}, \quad (4.23)$$

ki opisuje skaliranje za faktor s , zasuk za kot φ ter premik za x in y na ravnini. Ker želimo z matrikama ${}^W\mathbf{H}_R(t)$ in ${}^{R_t}\mathbf{H}_{R_{t+\tau}}$ opisati le premik in zasuk, zahtevamo da je velikost skalirnega faktorja enaka ena, torej $s = 1$.

Pri ocenjevanju matrike ${}^{R_t}\mathbf{H}_R(t)$ na podlagi enačbe (4.21) običajno ne dobimo zahtevane oblike (4.23). Rešitev zatorej poiščemo v smislu najboljšega približka. Omeniti velja, da to ni edini možni način razstavitve ocene homografije v smislu najboljšega približka, saj je pristop odvisen od izbire mere za določitev optimalnega približka. V kolikor je možno, je običajno še bolje, če lahko že med samim iskanjem ocene homografije upoštevamo posebno obliko le-te matrike — npr. oblike (4.23).

Pri uporabi preslikave homografije je treba biti previden. Zgodi se lahko, da se pri preslikavi točk, določene točke ne preslikajo v skladu z našimi pričakovanji. Problem nastane pri preslikavi točk, ki se nahajajo za slikovno ravnino in v resnici na sliki ne nastanejo, a preslikava homografije velja tudi za te točke, zato se pojavijo na sliki. Pri uporabi homografije za opis preslikav med slikami kamer moramo zato biti pazljivi. Pri ocenjevanju matrike ${}^{R_t}\mathbf{H}_R(t)$ na podlagi enačbe (4.21) običajno tudi ne dobimo zahtevane oblike (4.23). Rešitev lahko poiščemo v smislu najboljšega približka, a to ni vedno najboljši pristop. Običajno je bolje, da že med samim iskanjem ocene homografije upoštevamo posebno obliko le-te matrike — npr. oblike (4.23). Čeprav je možno na podlagi preslikave točk med slikami določiti relativni premik mobilnega sistema (na podlagi rezultata (4.21)), pa je zaradi omenjenih težav, relativni premik mobilnega sistema bolje oceniti neposredno v koordinatnem sistemu mobilnega sistema. Na podlagi predstavljenih preslikav je v poglavju 5.2.2 opisan postopek ocenjevanja lege mobilnega sistema z razširjenim Kalmanovim filtrom.

5 Vodenje mobilnega sistema po poti

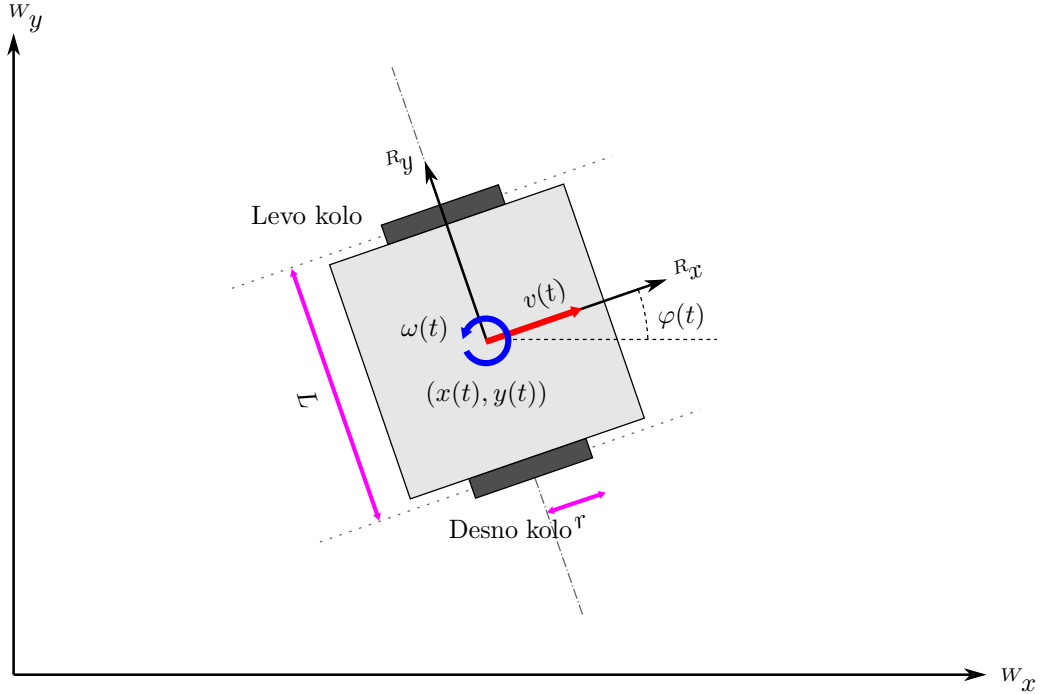
V tem delu je predstavljena izvedba vodenja dvokolesnega mobilnega sistema na diferencialni pogon po trajektoriji s strojnim vidom. Obravnavamo postavitev, kjer je kamera nameščena na mobilni sistem tako, da opazuje vozno površino pred mobilnim sistemom. Predpostavimo, da je vozna površina ravna in da vsebuje določen vzorec, ki se lahko ponavlja. Predstavljajmo si, da na sliki narišemo trajektorijo po kateri želimo, da se mobilni sistem pelje. V tem poglavju je prikazana izvedba regulatorja za vodenje po trajektoriji. V nadaljevanju je predstavljena izvedba ocenjevanja lege mobilnega sistema v prostoru s strojnim vidom in izračun regulacijskega pogreška s slike.

5.1 Kinematični model mobilnega sistema

Kinematični model dvokolesnega mobilnega sistema na diferencialni pogon je prikazan na sliki 5.1. Sistem je *neholonomičen*, saj ima eno neintegrabilno omejitev:

$$\mathbf{A}(\mathbf{q})\dot{\mathbf{q}} = \dot{x} \sin \varphi - \dot{y} \cos \varphi, \quad (5.1)$$

ki sledi iz predpostavke, da se mobilni sistem ne more gibati prečno glede na smer vrtenja koles. Matrika $\mathbf{A}(\mathbf{q})$ v enačbi (5.1) opisuje omejitev sistema, ki je podan v posplošenih koordinatah $\mathbf{q}^T = [x \ y \ \varphi]$. Če izrazimo vse možne hitrosti mobilnega sistema kot linearno kombinacijo vektorjev $\mathbf{s}_i(\mathbf{q})$, ki pripadajo ničelnemu prostoru



Slika 5.1: Model dvokolesnega mobilnega sistema na diferencialni pogon.

matrike $\mathbf{A}(\mathbf{q})$, pridemo do t. i. *kinematičnega modela prvega reda* [107]:

$$\dot{\mathbf{q}}(t) = \begin{bmatrix} \mathbf{s}_1 & \mathbf{s}_2 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix} = \begin{bmatrix} \cos(\varphi(t)) & 0 \\ \sin(\varphi(t)) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v(t) \\ \omega(t) \end{bmatrix}, \quad (5.2)$$

kjer je $v(t)$ tangencialna in $\omega(t)$ kotna hitrost mobilnega sistema. Model (5.2) dejansko opisuje kinematični model enokolesnika, saj model sovpada z modelom diferencialnega pogona v točki na središču osi med kolesoma. Model (5.2) lahko zapišemo v obliki, kjer kot vhodna signala nastopata kotni hitrosti levega kolesa $\omega_L(t)$ in desnega kolesa $\omega_D(t)$:

$$\dot{\mathbf{q}}(t) = \begin{bmatrix} \frac{R(\omega_D(t) + \omega_L(t))}{2} \cos \varphi(t) \\ \frac{R(\omega_D(t) + \omega_L(t))}{2} \sin \varphi(t) \\ \frac{R(\omega_D(t) - \omega_L(t))}{L} \end{bmatrix}. \quad (5.3)$$

Ob znanem radiju kolesa R in razdalji med kolesoma L sta kotni hitrosti levega in desnega kolesa mobilnega sistema povezani s tangencialno in kotno hitrostjo

mobilnega sistema z relacijo:

$$\omega_L(t) = \frac{1}{R} \left(v - \frac{L\omega}{2} \right), \quad \omega_D(t) = \frac{1}{R} \left(v + \frac{L\omega}{2} \right). \quad (5.4)$$

Obstajajo tudi drugačne predstavitve kinematičnega modela mobilnega sistema, kot je npr. *kinematični model diferencialnega pogona drugega reda* [107]. Za potrebe načrtovanja regulatorjev za vodenje mobilnih sistemov, so bile razvite tudi posebne oblike kinematičnega modela [105].

5.2 Ocenjevanje stanj

Pri vožnji mobilnega sistema po okolju, lahko njegovo lego določimo na podlagi meritev zasuka koles z enkoderji s t. i. postopkom odometrije. Postopek merjenja lege z odometrijo je zaradi integracije relativnih premikov podvržen lezenju. Z enkoderji ne moremo zaznati zdrsov koles, kar le še poveča napako v ocenjeni legi s pomočjo odometrije. Za merjenje zasukov koles se običajno uporabljajo inkrementalni enkoderji. Uporaba odometrije je izredno pogosta pri izvedbi avtonomnih mobilnih sistemov, a se zaradi omenjenih slabosti skoraj vedno uporablja v kombinaciji z drugimi senzorji za določanje lege. Postopek odometrije tako običajno srečamo v različnih izvedbah združevanja informacije s senzorjev, kjer se odometrija uporabi za napovedovanje stanj: določanje lege mobilnega sistema z razširjenim Kalmanovim filtrom [123], z Bayesovim filtrom [160], s filtrom delcev [161] itd. V nadaljevanju tega poglavja je predstavljen postopek določevanja lege mobilnega sistema s strojnim vidom, kjer je tudi uporabljen predikcija na podlagi odometrije. Natančneje, prikazan je postopek določevanja lege s t. i. pristopom *vizualne odometrije* [159, 162–165]. Kot velja za kolesno odometrijo, je tudi postopek vizualne odometrije podvržen lezenju, a je pri primerni izvedbi vizualne odometrije učinek neprimerljivo manjši [162]. Postopek ni občutljiv na zdrs koles, ki so glavni vir velikih napak pri običajni kolesni odometriji. Algoritem vizualne odometrije se od postopka *vizualne lokalizacije in gradnje zemljevida* [166] razlikuje po tem, da med samim postopkom ne poskušamo zagotoviti globalno optimalne rešitve. Pristop vizualne odometrije tako ne zna rešiti *problema zaključene poti*. Čeprav je pristop vizualne odometrije podvržen lezenju, pa je računska zahtevnost algoritmov vizualne odometrije precej učinkovitejša od algoritmov vizualne sočasne lokalizacije in gradnje zemljevida. Ker ocenjevanje lege ni

podvrženo integraciji napak zaradi zdrsov koles, je uporaba vizualne odometrije v aplikacijah vodenja mobilnih sistemov zelo aktualna. Sistem za izvedbo vizualne odometrije smo zasnovali za primer, ko se mobilni sistem giblje po ravni podlagi, ki jo opazujemo s kamero.

Lego mobilnega sistema lahko ocenjujemo le na podlagi primerjanja slikovnih značilk, lahko pa uporabimo še dodatne senzorje za merjenje premika (npr. enkoderje na kolesih mobilnega sistema ali neposredni kinematični model mobilnega sistema). Z uporabo kolesne odometrije lahko izboljšamo računsko učinkovitost iskanja parov med značilkami, saj lahko iskanje omejimo le na okolico, ki jo predvidimo na osnovi kolesne odometrije. Takšen pristop tudi izboljša ujemanje parov, še posebej v okoljih, kjer se določeni vzorci pogosto ponavljajo. Obdelava slike je računsko zahtevna operacija, obenem je omejena tudi frekvenca zajemanja slik, kar omejuje razpoložljivo frekvenco vzorčenja in vnaša tudi zakasnitve. S tega stališča je zanimiva uporaba kolesne odometrije, ki jo običajno lahko izvedemo z relativno visoko frekvenco. Vizualno in kolesno odometrijo lahko združimo v enotno oceno z uporabo metodologije združevanja informacij s senzorjev. V nadaljevanju bo predstavljen postopek, ki temelji na uporabi razširjenega Kalmanovega filtra.

5.2.1 Ocenjevanje lege na podlagi vizualne in kolesne odometrije

V tem poglavju se dotaknemo problema lokalizacije in sočasne gradnje zemljevida okolja. Obravnava je omejena na algoritme za lokalizacijo v neprirejenem strukturiranem zaprtem okolju. V nadaljevanju je predstavljen algoritem za lokalizacijo, ki temelji na uporabi razširjenega Kalmanovega filtra. Predstavljeni postopek lokalizacije je uporaben v splošnem primeru, ko s kamero na mobilnem sistemu opazujemo ravno površino po kateri se mobilni sistem giblje, ali pa je opazovana površina vzporedna z ravnino gibanja mobilnega sistema. Postavitev kamere na mobilnem sistemu je poljubna, če le zagotovi primerno vidno polje. Pri predstavljenem postopku poleg slikovne informacije uporabimo informacijo o relativnih premikih koles, ki nam služi za izračun kolesne odometrije.

Sistem zasnujemo na sledeč način. Razširjeni Kalmanov filter je ravno tako kot Kalmanov filter oz. splošni Bayesov filter sestavljen iz dveh delov: iz predikcijskega in korekcijskega dela. V našem primeru na podlagi znanega kinematičnega modela mobilnega sistema izvajamo predikcijski korak v opazovalniku stanj. Na

podlagi informacije s slike pa nato ocenjeno lego mobilnega sistema popravimo oz. izvedemo korekcijski korak v Kalmanovem filtru. Predpostavimo, da odčitavamo podatke s senzorjem z ekvidistančnim časom vzorčenja. Frekvenca zajemanja slike je omejena in strojna obdelava slik je časovno zahtevna operacija, medtem ko je frekvenca merjenja relativnih zasukov koles dokaj visoka. Kalmanov filter lahko zasnujemo tako, da izvajamo predikcijski korak z drugačno (višjo) frekvenco kot korekcijski korak.

V poglavju o strojnem vidu je opisano, kako lahko s pomočjo algoritma RANSAC na robusten način poiščemo pare točk med slikami. Tu je sedaj predstavljena nadgrajena različica algoritma za robustno iskanje parov in ocenjevanje lege kamere [164], ki nam na učinkovit način omogoča določevanje lege mobilnega sistema v realnem času. Algoritem se uporablja za t. i. slikovno/vizualno odometrijo [162, 167], ko le na podlagi slikovne informacije določimo relativno gibanje kamere v prostoru. Vizualna odometrija sodi med algoritme znane pod imenom angl. „*Structure from Motion — SfM*“, ki na podlagi gibanja kamere po prostoru ocenijo tako strukturo objektov v trirazsežnem prostoru kot tudi gibanje kamere po prostoru. Večina teh algoritmov je računsko izredno zahtevna in lahko običajno delujejo le v nesprotnem načinu. Algoritmi za vizualno odometrijo pa so običajno izdelani za sprotni način in so prirejeni za delovanje na zaporednem nizu slik. V nadaljevanju je prikazano, kako izvedemo vizualno odometrijo za naš primer vodenja mobilnega sistema na osnovi slike.

5.2.2 Opazovalnik stanj

Algoritem je sestavljen iz več korakov, ki v osnovi spominjajo na določene korake bodisi iz razširjenega Kalmanovega filtra bodisi iz algoritma RANSAC. Algoritem je zasnovan za diskretne sisteme s konstantnim časom vzorčenja T . Pri obravnavi upoštevamo, da so določeni signali lahko zakasneni. Zaradi lažje obravnave predpostavimo, da so signali zakasneni kvečjemu za celoštevilski mnogokratnik časa vzorčenja, torej $T_d = dT$, $d \in \mathbb{N}$. Upoštevamo tudi, da so določeni signali lahko vzorčeni z nižjo frekvenco, ki je — zopet zaradi enostavnejše obravnave — enaka celoštevilskemu faktorju signala z najvišjo frekvenco — le-ta je $\frac{1}{T}$ —, torej $\frac{1}{T_p} = \frac{1}{pT}$, $p \in \mathbb{N}$.

Predpostavimo, da naš sistem podaja naslednji sistem enačb:

$$\mathbf{x}(k+1) = \mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)), \quad (5.5)$$

$$\mathbf{y}(k) = \mathbf{h}(\mathbf{x}(k), \mathbf{v}(k)), \quad (5.6)$$

kjer so stanja sistema v trenutku k vsebovana v vektorju $\mathbf{x}(k) \in \mathbb{R}^n$, vektor $\mathbf{u}(k) \in \mathbb{R}^m$ predstavlja vhode v sistem, funkcija $\mathbf{f}(\cdot, \cdot, \cdot)$ pa podaja model prehajanja med stanji sistema. Vektor $\mathbf{w}(k)$ predstavlja beli šum s srednjo vrednostjo nič in kovariančno matriko $\mathbf{Q}(k)$ na stanjih, torej $\mathbf{w}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{Q}(k))$. Vektor $\mathbf{y}(k) \in \mathbb{R}^l$ predstavlja vektor merjenih signalov, ki so odvisni od stanj sistema, kot je podano v izhodni funkciji $\mathbf{h}(\cdot, \cdot)$. Vektor $\mathbf{v}(k)$ predstavlja beli šum na izhodu, ki ima srednjo vrednost enako nič in kovariančno matriko $\mathbf{R}(k)$, torej $\mathbf{v}(k) \sim \mathcal{N}(\mathbf{0}, \mathbf{R}(k))$. Na podlagi meritev v preteklih trenutkih želimo oceniti stanje sistema v trenutku k , torej želimo določiti oceno vektorja stanj $\mathbf{x}(k|k)$, tako da je ocena nepristranska in je kovarianca napake ocene minimalna. Kot je to običajno za Kalmanov filter, podamo algoritem v rekurzivni obliki, pri tem da posodobimo stanja sistema le v trenutku, ko je na voljo nova meritev izhodov sistema. Predikcije stanj izvajamo z ekvidistančnim časom vzorčenja, $t = kT$.

V našem primeru v vektorju stanj $\mathbf{x}^T(k)$ zberemo lego mobilnega sistema v svetovnem koordinatnem sistemu $(x(k), y(k), \varphi(k))$ in položaje točk v svetovnem koordinatnem sistemu $(^W p_{x,i}(k), ^W p_{y,i}(k))$:

$$\mathbf{x}^T(k) = \begin{bmatrix} x(k) & y(k) & \varphi(k) & \dots & ^W p_{x,i}(k) & ^W p_{y,i}(k) & \dots \end{bmatrix}. \quad (5.7)$$

Število točk, ki jih vzdržujemo v vektorju stanj, je časovno spremenljivo, saj med delovanjem opazovalnika stanj dodajamo nove vidne točke in odstranjujemo stare točke, ki že dolgo niso bile vidne. Vhodni vektor sestavljata tangencialna in kotna hitrost mobilnega sistema:

$$\mathbf{u}^T(k) = \begin{bmatrix} v(k) & \omega(k) \end{bmatrix}. \quad (5.8)$$

V izhodnem vektorju meritev se nahajajo izmerjeni položaji trenutno vidnih točk v lokalnem koordinatnem sistemu:

$$\mathbf{y}^T(k) = \begin{bmatrix} \dots & ^R p_{x,i}(k) & ^R p_{y,i}(k) & \dots \end{bmatrix}. \quad (5.9)$$

Pri tem velja poudariti, da je velikost izhodnega vektorja spremenljiva, saj niso vse točke venomer vidne v lokalnem koordinatnem sistemu mobilnega sistema. Predpostavimo, da model sistema opisuje naslednji sistem enačb:

$$\mathbf{f}(\mathbf{x}(k), \mathbf{u}(k), \mathbf{w}(k)) = \begin{bmatrix} x(k) + (v(k) + w_v(k))T \cos(\varphi(k)) \\ y(k) + (v(k) + w_v(k))T \sin(\varphi(k)) \\ \varphi(k) + (\omega(k) + w_\omega(k))T \\ \vdots \\ {}^W p_{x,i}(k) \\ {}^W p_{y,i}(k) \\ \vdots \end{bmatrix}, \quad (5.10)$$

kjer je $\mathbf{w}^T(k) = [w_v(k) \ w_\omega(k)]$ vektor šuma na vhodu (šum translatorne in kotne hitrosti). Predpostavimo, da izhodni model meritev opisuje sledeč sistem enačb:

$$\mathbf{h}(\mathbf{x}(k), \mathbf{v}(k)) = \begin{bmatrix} \vdots \\ \cos \varphi(k)({}^W p_{x,i}(k) - x(k)) + \sin \varphi(k)({}^W p_{y,i}(k) - y(k)) + v_{x,i}(k) \\ -\sin \varphi(k)({}^W p_{x,i}(k) - x(k)) + \cos \varphi(k)({}^W p_{y,i}(k) - y(k)) + v_{y,i}(k) \\ \vdots \end{bmatrix}, \quad (5.11)$$

kjer je $\mathbf{v}^T(k) = [\dots v_{x,i}(k) \ v_{y,i}(k) \dots]$.

Predikcijski korak

Začnemo s predikcijskim korakom. Po običajnem postopku predikcije, kot jo poznamo v razširjenem Kalmanovem filtru naredimo predikcijo stanj $\mathbf{x}(k|k-1)$ in pripadajoče kovariančne matrike $\mathbf{P}(k|k-1) = \text{cov}\{\mathbf{x}(k) - \mathbf{x}(k|k-1)\}$:

$$\mathbf{x}(k|k-1) = \mathbf{f}(\mathbf{x}(k-1|k-1), \mathbf{u}(k-1), \mathbf{0}), \quad (5.12)$$

$$\mathbf{P}(k|k-1) = \mathbf{A}(k)\mathbf{P}(k-1|k-1)\mathbf{A}^T(k) + \mathbf{W}(k)\mathbf{Q}(k-1)\mathbf{W}^T(k), \quad (5.13)$$

kjer je $\mathbf{x}(k-1|k-1)$ ocena stanj v trenutku $k-1$, matrika $\mathbf{P}(k-1|k-1) = \text{cov}\{\mathbf{x}(k-1) - \mathbf{x}(k-1|k-1)\}$ je kovariančna matrika napake ocene, matrika $\mathbf{Q}(k-1)$ je kovariančna matrika šuma na stanjih. Vektor $\mathbf{x}(k|k-1)$ predstavlja predikcijo stanj, ki ji pripada kovariančna matrika $\mathbf{P}(k|k-1)$. V enačbi (5.13)

sta Jacobijevi matriki $\mathbf{A}(k) \in \mathbb{R}^n \times \mathbb{R}^n$ in $\mathbf{W}(k) \in \mathbb{R}^n \times \mathbb{R}^n$ definirani, kot sledi:

$$\mathbf{A}(k) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \right|_{(\mathbf{x}(k-1|k-1), \mathbf{u}(k-1), \mathbf{0})}, \quad (5.14)$$

$$\mathbf{W}(k) = \left. \frac{\partial \mathbf{f}}{\partial \mathbf{w}} \right|_{(\mathbf{x}(k-1|k-1), \mathbf{u}(k-1), \mathbf{0})}. \quad (5.15)$$

Iskanje parov

Ko je na voljo nova slika, s primernim postopkom strojnega vida poiščemo pare točk med dvema slikama, ki ustrezajo isti točki v prostoru. Postopek zaznavanja, opisovanja in iskanja parov značilk je opisan v poglavju o strojnem vidu. Predvsem v primeru, ko imamo opravka s ponavljajočimi se vzorci, lahko pride do velikega števila napačnih parov značilk. Problem lahko rešimo, tako da opravimo iskanje le v omejeni okolici, ki jo določimo na podlagi predikcije meritev:

$$\mathbf{y}(k|k) = \mathbf{h}(\mathbf{x}(k|k-1), \mathbf{0}), \quad (5.16)$$

$$\mathbf{S}(k|k) = \mathbf{C}(k)\mathbf{P}(k|k-1)\mathbf{C}^T(k) + \mathbf{V}(k)\mathbf{R}(k)\mathbf{V}^T(k), \quad (5.17)$$

kjer sta Jacobijevi matriki $\mathbf{C} \in \mathbb{R}^n \times \mathbb{R}^n$ in $\mathbf{V} \in \mathbb{R}^n \times \mathbb{R}^n$ določeni, kot sledi:

$$\mathbf{C}(k) = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{x}} \right|_{(\mathbf{x}(k|k-1), \mathbf{0})}, \quad (5.18)$$

$$\mathbf{V}(k) = \left. \frac{\partial \mathbf{h}}{\partial \mathbf{v}} \right|_{(\mathbf{x}(k|k-1), \mathbf{0})}. \quad (5.19)$$

Uporaba predikcije običajno tudi olajša iskanje parov značilk, saj nam omogoča, da iskanje omejimo le na okolico, ki je določena s kovariančno matriko $\mathbf{S}(k|k) = \text{cov}\{(\mathbf{y}(k) - \mathbf{y}(k|k))\}$ (rešitve iščemo v okolici z visoko stopnjo verjetnosti, običajno 0,99). Pare značilk med trenutno in predhodno sliko zapišemo v matriki $\mathbf{M} \in \mathbb{N}^2 \times \mathbb{N}^M$, kjer stolpec $\mathbf{m}_i \in \mathbf{M}$ sestoji iz indeksov značilk, ki sestavljata par $i = 1, 2, \dots, M$ ($M \geq 0$ je število najdenih parov).

Postavitev in preverjanje hipoteze

Sledi postavitev in preverjanje hipoteze. Hipotezo postavimo na sledeč način. Izmed vseh najdenih parov $\mathbf{M}$ naključno izberemo par $\mathbf{m}_p$, ki ga sestavljata značilki $\mathbf{y}_i(k-1)$ in $\mathbf{y}_j(k)$.

Matrika (5.18) ima obliko $\mathbf{C}^T(k) = [\mathbf{C}_0^T(k) \ \mathbf{C}_1^T(k) \ \dots \ \mathbf{C}_S^T(k)]$, kjer je $\mathbf{C}_i(k) \in \mathbb{R}^l \times \mathbb{R}^N$.

$$\mathbf{y}_i(k) = \mathbf{C}_i(k)\mathbf{x}(k), \quad (5.20)$$

Napravimo hipotezo ocene stanj na podlagi enega vzorca meritev:

$$\mathbf{x}_h(k|k) = \mathbf{x}(k|k-1) + \mathbf{L}(k)(\mathbf{y}_j(k) - \mathbf{C}_i(k)\mathbf{x}(k|k-1)), \quad (5.21)$$

$$\mathbf{L}(k) = \mathbf{P}(k|k-1)\mathbf{C}_i^T(k)(\mathbf{C}_i(k)\mathbf{P}(k|k-1)\mathbf{C}_i^T(k) + \mathbf{R}_i(k))^{-1}. \quad (5.22)$$

Veljavnost hipoteze ocene $\mathbf{x}_h(k|k)$ preverimo tako, da izračunamo, kako dobro se postavljena hipoteza sklada z vsemi vzorci:

$$E_h = \|\mathbf{y}_h(k|k) - \mathbf{y}(k)\|, \quad (5.23)$$

$$\mathbf{y}_h(k|k) = \mathbf{C}(k)\mathbf{x}_h(k|k). \quad (5.24)$$

Če je napaka hipoteze E_h manjša od do tedaj najmanjše napake, izberemo trenutno hipotezo za najboljšo. V tem primeru ponovno izračunamo, koliko ponovitev moramo izvesti, da dosežemo določeno verjetnost p , da je najmanj ena izmed hipotez pravilna:

$$H = \frac{\log(1-p)}{\log(1-(1-\epsilon)^\eta)}, \quad (5.25)$$

kjer je ϵ razmerje med številom pripadajočih vzorcev in številom vseh vzorcev in η minimalno število vzorcev, ki jih potrebujemo, da ocenimo model. Enačba (5.25) velja splošno za algoritem RANSAC. V našem primeru je $\eta = 1$.

Korekcijski korak

Množico vzorcev, ki najbolj opisuje dane podatke, uporabimo sedaj, da dokončno izračunamo korekcijsko oceno stanj in pripadajočo kovariančno matriko napake ocene:

$$\mathbf{x}(k|k) = \mathbf{x}(k|k-1) + \mathbf{L}(k)(\mathbf{y}(k) - \mathbf{C}(k)\mathbf{x}(k|k-1)), \quad (5.26)$$

$$\mathbf{L}(k) = \mathbf{P}(k|k-1)\mathbf{C}^T(k)(\mathbf{C}(k)\mathbf{P}(k|k-1)\mathbf{C}^T(k) + \mathbf{R}(k))^{-1}, \quad (5.27)$$

$$\mathbf{P}(k|k) = (\mathbf{I} - \mathbf{L}(k)\mathbf{C}(k))\mathbf{P}(k|k-1) \quad (5.28)$$

Dodajanje in odstranjevanje stanj

V korekcijskem koraku posodobimo ocenjena stanja na podlagi odkritih parov točk med slikama. Po končanem koraku korekcije pa vektor stanj razširimo še z novimi vzorci, ki nimajo para na prejšnji sliki. Takšen način omogoča izvedbo vizualne odometrije na daljše razdalje, a prinaša tudi težave, saj se pri tem dimenzije sistema neomejeno povečujejo. Za delovanje takšnega sistema je zato nujno vpeljati dodatne mehanizme, ki odstranjujejo pare, ki niso potrebni. Takšen mehanizem lahko zasnujemo na način, da preverjamo starost točk v vektorju stanj in točke, ki že dolgo niso bile vidne odstranimo. Pri tem odstranimo elemente iz vektorja stanj, kar zahteva tudi odstranitev ustreznih vrstic in stolpcev v kovariančnih matrikah.

5.3 Vodenje po trajektoriji

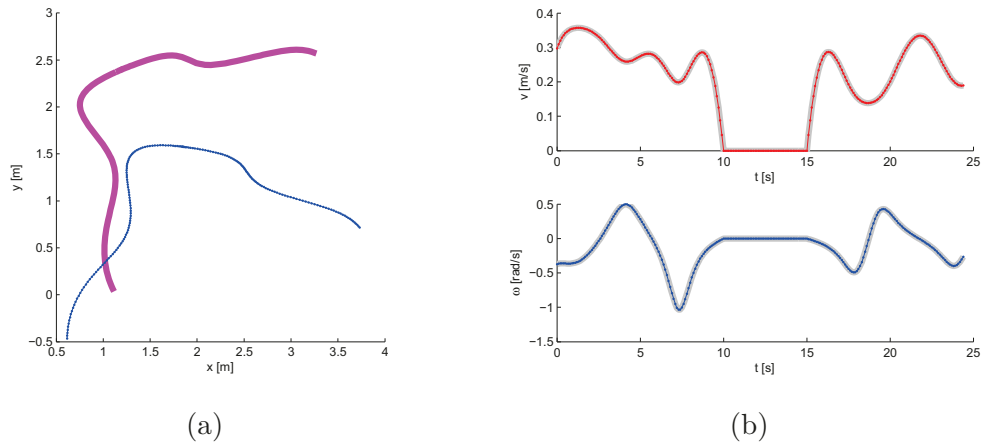
Vodenje po trajektoriji zasnujemo na podlagi kinematičnega modela mobilnega sistema. Kinematični model mobilnega sistema na diferencialni pogon (5.2) je t. i. *gladek sistem* (angl. „*flat system*”) [6], z gladkima izhodoma x in y . Ta lastnost zagotavlja obstoj zveznih regulirnih signalov $v_r(t)$ in $\omega_r(t)$, ki v idealnem primeru zagotovita sledenje gladki referenčni trajektoriji brez kršitev kinematične omejitve (5.1) [168]. Odprtozančna signala $v_r(t)$ in $\omega_r(t)$ lahko izračunamo iz vsaj dvakrat zvezno odvedljive parametrične krivulje $(x_r(t), y_r(t))$:

$$v_r(t) = \sqrt{\dot{x}_r^2(t) + \dot{y}_r^2(t)}, \quad (5.29)$$

$$\omega_r(t) = \frac{\dot{x}_r(t)\ddot{y}_r(t) - \ddot{x}_r(t)\dot{y}_r(t)}{\dot{x}_r^2(t) + \dot{y}_r^2(t)}. \quad (5.30)$$

To omogoča, da iz primerne referenčne krivulje neposredno določimo referenčni krmilni hitrosti. Izkaže se, da so za načrtovanje poti še posebej primerne Bernstein-Bézierjeve parametrične krivulje, kar je predstavljeno v poglavju 6. Primer odprtozančnega vodenja mobilnega sistema s krmilnima signaloma (5.29) in (5.30) je prikazan na sliki 5.2. Regulacijski algoritem tako zasnujemo po principu dvoprostostne regulacije [169], kjer regulirno akcijo zasnujemo kot vsoto odprtozančnega krmilnega signala in povratnozančnega regulacijskega signala, ki temelji na merjenju oz. ocenjevanju pogreška. Takšen pristop je običajen pri izvedbi sledenja

trajektoriji [6, 16, 105]. V nadaljevanju je v poglavju 5.4 prikazan pristop izvedbe povratnozančne regulacije s prediktivnim regulatorjem [16, 108].



Slika 5.2: Odprtozančno vodenje pri neničelnem začetnem pogrešku ($e_x(0) = 0,5$ m, $e_y(0) = 0,5$ m in $e_\varphi(0) = 0,5$ rad). (a) Referenčna (krepko vijolično) in dejanska trajektorija ter (b) krmilna signala.

Čeprav postane pri ničelni tangencialni hitrosti (5.29) izraz (5.30) za kotno hitrost nedoločen, lahko kotno hitrost v tem primeru določimo s pomočjo L'Hôpitalovega pravila, kar pa zahteva, da je referenčna krivulja primerno velikokrat zvezno odvedljiva ali povsem gladka. V poglavju 6 so posebej obravnavani tudi primeri, ko tangencialna hitrost v določenem trenutku doseže ničelno tangencialno hitrost — takšen je tudi primer na sliki 5.2b. Izvedbe regulatorjev za zaprtozančno vodenje dvokolesnih mobilnih sistemov po trajektoriji običajno zahtevajo, da referenčna tangencialna hitrost ni enaka nič [6, 16, 105], sicer postane sistem nevodljiv. Pri načrtovanju regulacije moramo tako sprejeti določen kompromis: ali se zanašamo le na odprtozančno vodenje ali se odpovemo sledenju referenčni trajektoriji. V poglavju 5.4.1 je predstavljen primer prediktivnega regulatorja, ki za vodljivost načeloma zahteva bodisi neničelno tangencialno hitrost bodisi neničelno kotno hitrost [16]. V nadaljevanju pa je v poglavju 5.4.2 izpeljan še primer regulatorja za sledenje trajektoriji, ki nima takšne omejitve.

5.4 Prediktivni regulator na osnovi modela pogreška

Oglejmo si najprej splošno zgradbo prediktivnega regulatorja na osnovi modela pogreška. V nadaljevanju je predstavljeni prediktivni regulator uporabljen za izvedbo vodenja mobilnega sistema po trajektoriji na več različnih načinov — glede na uporabljeni model pogreška. Predpostavimo, da imamo na voljo model pogreška, ki je zapisan v diskretnem prostoru stanj:

$$\boldsymbol{\epsilon}(k+1) = \mathbf{A}(k)\boldsymbol{\epsilon}(k) + \mathbf{B}(k)\mathbf{u}_B(k), \quad (5.31)$$

$$\mathbf{e}(k) = \mathbf{C}(k)\boldsymbol{\epsilon}(k). \quad (5.32)$$

Prediktivni regulator zasnujemo kot optimizacijski problem, kjer iščemo optimalne vrednosti regulirnih signalov, ki minimizirajo določeno kriterijsko funkcijo na časovno omejenem napovednem horizontu $h > 0$. V [16, 108] je predlagana uporaba naslednje kriterijske funkcije za izvedbo prediktivnega vodenja na osnovi modela pogreška:

$$\begin{aligned} J(\mathbf{u}_B, k) = & \sum_{i=0}^{h-1} \boldsymbol{\epsilon}^T(k, i+1) \mathbf{Q}(i+1) \boldsymbol{\epsilon}(k, i+1) + \\ & + \mathbf{u}_B^T(k+i|k) \mathbf{R}(i) \mathbf{u}_B(k+i|k), \end{aligned} \quad (5.33)$$

kjer podaja vektor $\boldsymbol{\epsilon}(k, i) = \boldsymbol{\epsilon}_r(k+i|k) - \boldsymbol{\epsilon}(k+i|k) \in \mathbb{R}^n$ razliko med želenim in ocenjenim pogreškom; vektor $\mathbf{u}_B(k+i|k) \in \mathbb{R}^m$ podaja regulirne signale. Matriki $\mathbf{Q}(i+1) \in \mathbb{R}^{n \times n}$ in $\mathbf{R}(i) \in \mathbb{R}^{m \times m}$ poskrbita za primerno obtežbo členov v kriterijski funkciji in sta pozitivno definitni za vsak $i = 0, 1, \dots, h-1$: $\mathbf{x}^T \mathbf{Q}(i+1) \mathbf{x} > 0$ in $\mathbf{y}^T \mathbf{R}(i) \mathbf{y} > 0$ za vsak $\mathbf{x} \neq \mathbf{0}$ in $\mathbf{y} \neq \mathbf{0}$.

V koraku k lahko na podlagi modela pogreška (5.31) ocenimo pogrešek v koraku $k+h$, če le poznamo trenutno stanje in regulirne signale do časovnega trenutka $k+h-1$:

$$\begin{aligned} \boldsymbol{\epsilon}(k+h|k) = & \boldsymbol{\Lambda}_h(k, 0) \boldsymbol{\epsilon}(k|k) + \\ & + \sum_{i=1}^{h-1} \boldsymbol{\Lambda}_h(k, i) \mathbf{B}(k+i-1|k) \mathbf{u}_B(k+i-1) + \\ & + \mathbf{B}(k+h-1|k) \mathbf{u}_B(k+h-1), \end{aligned} \quad (5.34)$$

kjer smo vpeljali $\boldsymbol{\Lambda}_h(k, i) = \mathbf{A}(k+h-1|k) \dots \mathbf{A}(k+i+1|k) \mathbf{A}(k+i|k)$. Če zberemo vsa stanja na prediktivnem horizontu h v razširjenem vektorju stanj

$\bar{\epsilon}(k) \in \mathbb{R}^{n \cdot h}$:

$$\bar{\epsilon}^T(k) = \begin{bmatrix} \epsilon^T(k+1|k) & \epsilon^T(k+2|k) & \dots & \epsilon^T(k+h|k) \end{bmatrix}, \quad (5.35)$$

in vse regulirne signale na prediktivnem intervalu h v razširjenem vhodnem vektorju $\bar{\mathbf{u}}_B(k) \in \mathbb{R}^{m \cdot h}$:

$$\bar{\mathbf{u}}_B^T(k) = \begin{bmatrix} \mathbf{u}_B^T(k|k) & \mathbf{u}_B^T(k+1|k) & \dots & \mathbf{u}_B^T(k+h-1|k) \end{bmatrix}, \quad (5.36)$$

lahko zapišemo t. i. razširjeni model pogreška:

$$\bar{\epsilon}(k) = \bar{\mathbf{A}}(k)\epsilon(k|k) + \bar{\mathbf{B}}(k)\bar{\mathbf{u}}_B(k). \quad (5.37)$$

Matriki $\bar{\mathbf{A}} \in \mathbb{R}^{n \cdot h} \times \mathbb{R}^n$ in $\bar{\mathbf{B}} \in \mathbb{R}^{n \cdot h} \times \mathbb{R}^{m \cdot h}$ izpeljemo iz modela pogreška (5.34):

$$\bar{\mathbf{A}}^T(k) = \begin{bmatrix} \mathbf{A}^T(k|k) & \mathbf{A}^T(k|k)\mathbf{A}^T(k+1|k) & \dots & \mathbf{\Lambda}_h^T(k,0) \end{bmatrix}, \quad (5.38)$$

$$\bar{\mathbf{B}}(k) = \begin{bmatrix} \mathbf{B}(k|k) & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{A}(k+1|k)\mathbf{B}(k|k) & \mathbf{B}(k+1|k) & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{\Lambda}_h(k,1)\mathbf{B}(k|k) & \mathbf{\Lambda}_h(k,2)\mathbf{B}(k+1|k) & \dots & \mathbf{B}(k+h-1|k) \end{bmatrix}. \quad (5.39)$$

Referenčne pogreške na napovednem intervalu h zberemo v razširjenem vektorju:

$$\bar{\epsilon}_r^T(k) = \begin{bmatrix} \epsilon_r^T(k+1|k) & \epsilon_r^T(k+2|k) & \dots & \epsilon_r^T(k+h|k) \end{bmatrix}. \quad (5.40)$$

Kriterijsko funkcijo (5.33) lahko sedaj prepišemo v matrično obliko:

$$J(\bar{\mathbf{u}}_B, k) = \bar{\epsilon}^T(k)\bar{\mathbf{Q}}\bar{\epsilon}(k) + \bar{\mathbf{u}}_B^T(k)\bar{\mathbf{R}}\bar{\mathbf{u}}_B(k), \quad (5.41)$$

kjer smo upoštevali:

$$\bar{\epsilon}(k) = \bar{\epsilon}_r(k) - \bar{\epsilon}(k), \quad (5.42)$$

$$\bar{\mathbf{Q}} = \bigoplus_{i=1}^h \mathbf{Q}(i), \quad \bar{\mathbf{R}} = \bigoplus_{i=0}^{h-1} \mathbf{R}(i). \quad (5.43)$$

Operator $\oplus$ ponazarja operacijo direktne vsote oz. bločno diagonalno razporeditev matrik. Pri načrtovanju regulatorja moramo določiti vektor razširjenega referenčnega pogreška $\bar{\epsilon}_r(k)$, kar lahko storimo na različne načine. V delu [16] je predlagana uporaba dinamičnega modela prvega reda, ki temelji na trenutni vrednosti pogreška:

$$\bar{\epsilon}_r(k) = \begin{bmatrix} \mathbf{A}_r^T & \mathbf{A}_r^{2T} & \dots & \mathbf{A}_r^{hT} \end{bmatrix}^T \epsilon(k|k), \quad (5.44)$$

kjer sistemska matrika $\mathbf{A}_r \in \mathbb{R}^n \times \mathbb{R}^n$ podaja dinamiko referenčnega modela pogreška. Možna bi bila tudi izbira $\bar{\epsilon}_r(k) = \mathbf{0}$, ki je posebna oblika (5.44), ko je $\mathbf{A}_r = \mathbf{0}$. Takšna izbira bi zahtevala hipno zmanjšanje pogreška, medtem ko z eksponentnim padanjem referenčnega pogreška dosežemo bolj gladko približevanje referenčni trajektoriji.

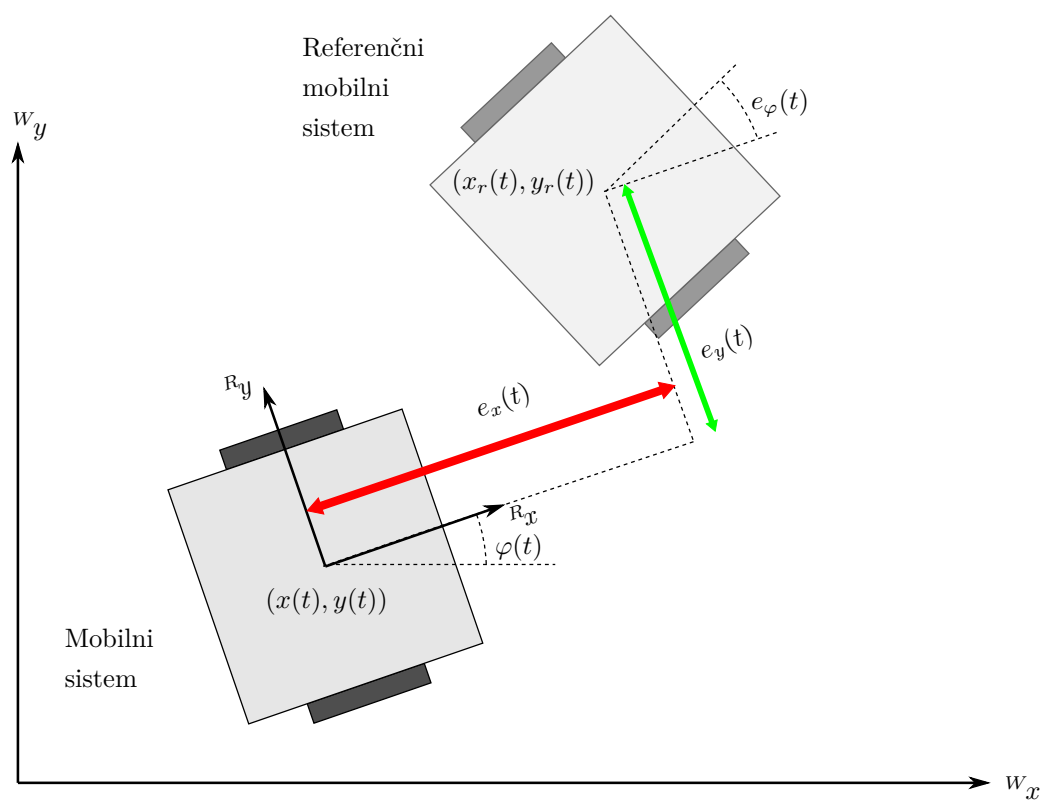
Optimalno vrednost regulirnega signala določimo tako, da poiščemo minimum kriterijske funkcije (5.41). Z iskanjem ničel odvoda kriterijske funkcije (5.41), $\frac{\partial J(\bar{\mathbf{u}}_B, k)}{\partial \bar{\mathbf{u}}_B(k)} = 0$, najdemo njeno kritično točko:

$$\bar{\mathbf{u}}_B(k) = \left(\bar{\mathbf{B}}^T(k) \bar{\mathbf{Q}} \bar{\mathbf{B}}(k) + \bar{\mathbf{R}} \right)^{-1} \bar{\mathbf{B}}^T(k) \bar{\mathbf{Q}} \left(\bar{\epsilon}_r(k) - \bar{\mathbf{A}}(k) \epsilon(k|k) \right). \quad (5.45)$$

Preverimo lahko, da je — glede na upoštevane predpostavke — najdena kritična točka tudi minimum kriterijske funkcije (5.41) oz. (5.33). Glede na strategijo prediktivnega vodenja, ki je znana tudi kot strategija *neulovljivega horizonta*, v trenutku k upoštevamo le prvih m optimalnih regulirnih vrednosti: $\mathbf{u}_B(k) = [\mathbf{I} \ \mathbf{0} \ \dots \ \mathbf{0}] \bar{\mathbf{u}}_B(k)$; in nato v naslednjem koraku celoten postopek izračuna optimalnih regulirnih vrednosti ponovimo. Dobra lastnost prikazane izvedbe prediktivnega regulatorja je, da so optimalne regulirne vrednosti analitično določljive, kar izloči potrebo po uporabi iterativnih numeričnih optimizacijskih postopkov, ki so običajno precej računsko zahtevni. To dejstvo nam omogoča, da regulirne signale izračunamo na računsko učinkovit način, kar olajša izvedbo regulatorja za delovanje v realnem času.

5.4.1 Model pogreška

Pogrešek za sledenje referenčnemu vozilu običajno podamo v lokalnem koordinatnem sistemu mobilnega sistema tako, kot je bilo predlagano v [16, 105]. Pogrešek



Slika 5.3: Lega referenčnega mobilnega sistema glede na mobilni sistem.

v svetovnem koordinatnem sistemu $\mathbf{q}_r - \mathbf{q}$ lahko zapišemo v koordinatnem sistemu mobilnega sistema (slika 5.3):

$$\mathbf{e} = \begin{bmatrix} e_x \\ e_y \\ e_\varphi \end{bmatrix} = \overbrace{\begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}}^{\mathbf{R}} (\mathbf{q}_r - \mathbf{q}). \quad (5.46)$$

Ob upoštevanju, da referenčno vozilo upošteva kinematični model mobilnega sistema (5.2), lahko odvod pogreška (5.46) zapišemo v naslednji obliki:

$$\dot{\mathbf{e}} = \mathbf{R}(\dot{\mathbf{q}}_r - \dot{\mathbf{q}}) + \omega \begin{bmatrix} e_y \\ -e_x \\ 0 \end{bmatrix} = \begin{bmatrix} v_r \cos e_\varphi - v + \omega e_y \\ v_r \sin e_\varphi - \omega e_x \\ \omega_r - \omega \end{bmatrix}. \quad (5.47)$$

Sistem (5.47) je nelinearni model pogreška, ki ga uporabimo za izvedbo prediktivnega regulatorja. Regulirni signal zasnujemo kot vsoto $\mathbf{u} = \mathbf{u}_F + \mathbf{u}_B$, kjer je $\mathbf{u}_B^T = [v_B \ \omega_B]$ zaprtozančni regulirni signal. Običajno uporabimo za krmilno akcijo zvezo $\mathbf{u}_F^T = [v_r \cos e_\varphi \ \omega_r]$ [16, 105]. Zapišemo lahko torej naslednjo verzijo pogreška:

$$\dot{\mathbf{e}} = \begin{bmatrix} (\omega_r + \omega_B)e_y - v_B \\ v_r \sin e_\varphi - (\omega_r + \omega_B)e_x \\ -\omega_B \end{bmatrix}. \quad (5.48)$$

Za izvedbo prediktivnega regulatorja (poglavje 5.4) potrebujemo linearni model pogreška. Model pogreška (5.47) lineariziramo okoli referenčne trajektorije ($e_x = 0$, $e_y = 0$, $e_\varphi = 0$, $v_B = 0$ in $\omega_B = 0$) [16]:

$$\dot{\mathbf{e}} = \begin{bmatrix} 0 & \omega_r & 0 \\ -\omega_r & 0 & v_r \\ 0 & 0 & 0 \end{bmatrix} \mathbf{e} + \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & -1 \end{bmatrix} \mathbf{u}_B. \quad (5.49)$$

Linearizacija modela pogreška okoli referenčne trajektorije ni edina možna, a je za majhne odmikne od referenčne trajektorije običajno dovolj dobra. Za izvedbo prediktivnega regulatorja (poglavje 5.4) pretvorimo model (5.49) iz zveznega prostora v diskretni prostor s časom vzorčenja T . Za majhne čase vzorčenja lahko uporabimo kar približek $\mathbf{A} = \mathbf{I} + \mathbf{A}_z T$ in $\mathbf{B} = \mathbf{B}_z T$ za pretvorbo zveznega sistema $\dot{\mathbf{e}}(t) = \mathbf{A}_z \mathbf{e}(t) + \mathbf{B}_z \mathbf{u}_B(t)$ v diskretni sistem $\mathbf{e}(k+1) = \mathbf{A} \mathbf{e}(k) + \mathbf{B} \mathbf{u}_B(k)$.

Vodljivost modela pogreška (5.49) lahko preverimo z izračunom vodljivostne matrike:

$$\begin{bmatrix} \mathbf{B} & \mathbf{AB} & \mathbf{A}^2\mathbf{B} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & 0 & \omega_r^2 & -\omega_r v_r \\ 0 & 0 & \omega_r & -v_r & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (5.50)$$

Na podlagi (5.50) lahko zaključimo, da je model pogreška (5.49) vodljiv, če je bodisi referenčna tangencialna hitrost različna od nič ($v_r \neq 0$) bodisi je referenčna kotna hitrost različna od nič ($\omega_r \neq 0$). Takšna obravnava velja le pri konstantnih referenčnih hitrostih ($v_r = \text{konst.}$ in $\omega_r = \text{konst.}$), sicer moramo preveriti vodljivost s kriteriji za časovno spremenljive sisteme [16]. Zaključimo lahko torej, da v primeru ničelnih referenčnih hitrosti ($v_r = 0$ ali $\omega_r = 0$) model pogreška (5.49) ni vodljiv. Model pogreška je torej primeren za izvedbo regulatorja za vodenje po trajektoriji, vendar ni hkrati primeren tudi za regulacijo lege, saj ne more izničiti vseh elementov v vektorju pogreška. Znano je, da sta vodenje po trajektoriji in regulacija lege dva ločena problema, ki ju običajno ne moremo razrešiti z istim regulatorjem [6, 16]. Regulatorji za sledenje trajektoriji običajno zahtevajo neničelno tangencialno hitrosti ($v_r \neq 0$) [16, 168]. Nekateri regulacijski algoritmi imajo pri majhnih tangencialnih hitrostih veliko ojačenje na kotni hitrosti, kar je še posebej nezaželeno, saj le-to privede do velikih oscilacij kotne hitrosti.

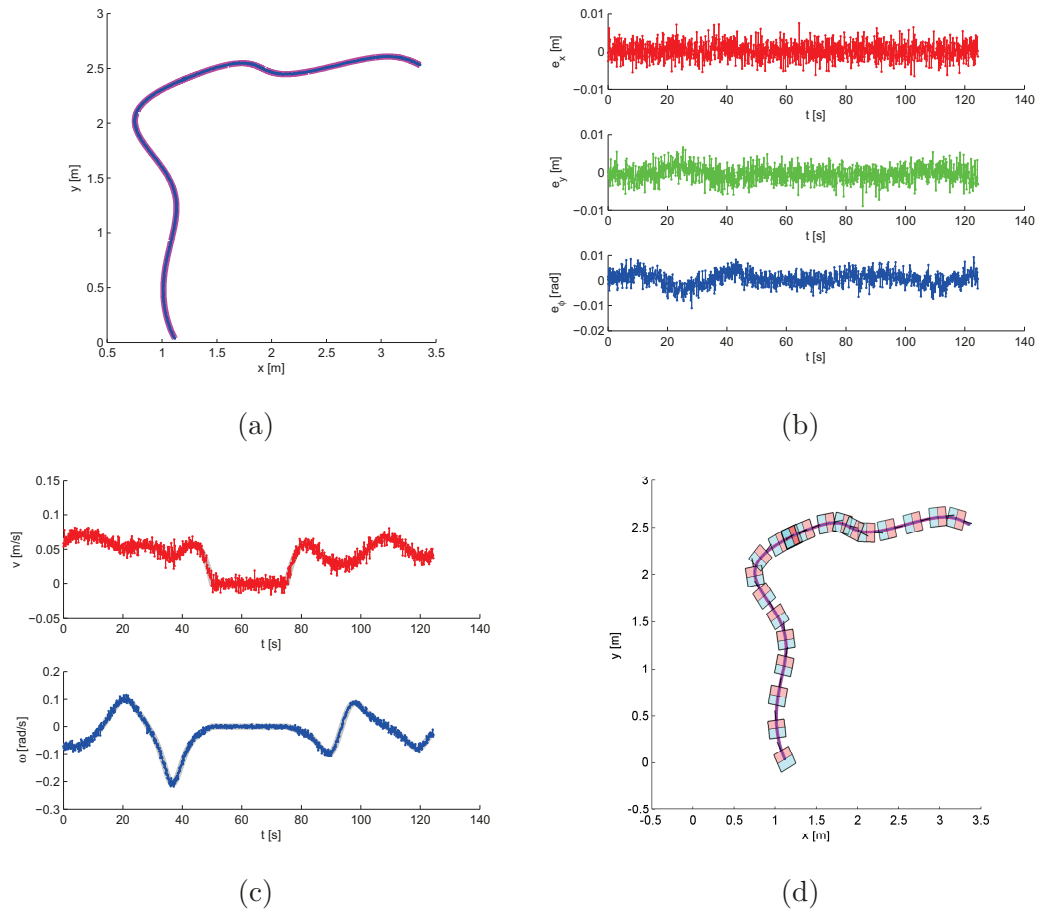
Na sliki 5.4 je predstavljeno delovanje prediktivnega regulatorja, ko začetna lega mobilnega sistema sovpada z referenčno lego in je izhod moten z normalno porazdeljenim belim šumom s standardno deviacijo $\sigma_{n,x} = \sigma_{n,y} = 0,002 \text{ m}$ in $\sigma_{n,\varphi} = 0,002 \text{ rad}$. Na slikah 5.5 in 5.6 sta prikazani situaciji pri neničelnem začetnem pogrešku ($e_x(0) = 0,5 \text{ m}$, $e_y(0) = 0,5 \text{ m}$ in $e_\varphi(0) = 0,5 \text{ rad}$), brez in s šumom na izhodu. V primeru na sliki 5.7 je podano še obnašanje regulatorja, ko ob času $t = 60 \text{ s}$ nastopi motnja na izhodu. Delovanje prediktivnega regulatorja za sledenje trajektoriji je z naborom cenilk ovrednoteno v tabeli 5.1¹. Izbrani so bili sledeči parametri prediktivnega regulatorja: $h = 5$, $\mathbf{Q} = \text{diag}([5, 50, 0,5]^T)$,

¹Definicije cenilk za izmero odstopanja x :

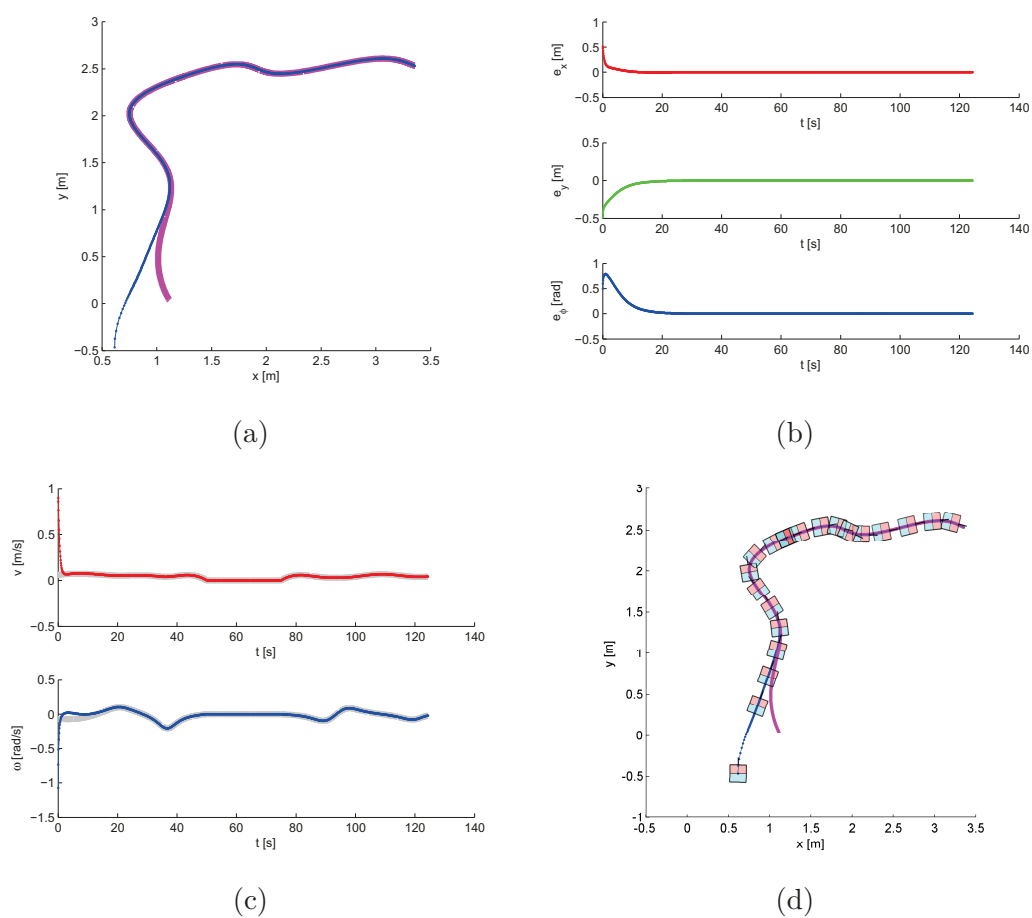
$$\begin{aligned} SSE(x) &= \sum_k x^2(k), & SAE(x) &= \sum_k \|x(k)\|, \\ TSSE(x) &= \sum_k t(k)x^2(k), & TSAE(x) &= \sum_k t(k)\|x(k)\|, \end{aligned}$$

$\mathbf{R} = \text{diag}([0,5, 0,1]^T)$ in $\mathbf{A}_r = \text{diag}([0,65, 0,65]^T)$. Predstavljeni primeri kažejo, da je s predstavljenim regulatorjem možno točno sledenje referenčni trajektoriji. Izkaže se, da prediktivni regulator na osnovi predstavljenega modela pogreška ne povzroča velikih oscilacij kotne hitrosti in je stabilen, tudi ko padeta referenčni hitrosti na nič. A s slike 5.7 je razvidno, da regulator ni zmožen odpraviti pogreška (5.49), ko sta obe referenčni hitrosti enaki nič. To potrjuje ugotovitve o vodljivosti modela pogreška.

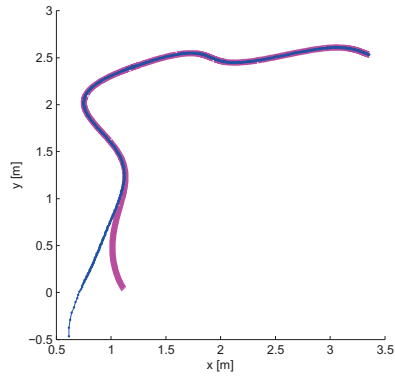
Na sliki 5.8 je prikazano sledenje referenčni trajektoriji s prediktivnim regulatorjem, ki je bilo izvedeno na realnem mobilnem sistemu Pioneer 3-AT. Za parametre regulatorja so bile izbrane enake vrednosti kot v simulacijskih primerih na slikah 5.4 do 5.7. Regulirni hitrosti sta bili v tem primeru omejeni: $|v| < 0,2 \text{ m/s}$ in $|\omega| < 1 \text{ rad/s}$. Delovanje regulatorja je ovrednoteno v tabeli 5.2.



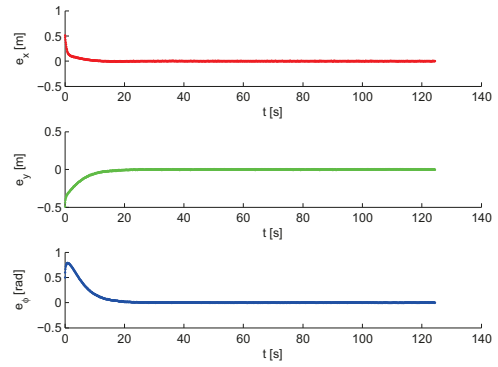
Slika 5.4: Sledenje trajektoriji s prediktivnim regulatorjem pri ničelnem začetnem pogrešku in normalno porazdeljenemu belemu šumu s standardno deviacijo $\sigma_{n,x} = \sigma_{n,y} = 0,002$ m in $\sigma_{n,\varphi} = 0,002$ rad na izhodu. (a) Referenčna (krepko) in dejanska trajektorija, (b) pogrešek, (c) regulirna signala in (d) lege mobilnega sistema vsakih polnih pet sekund.



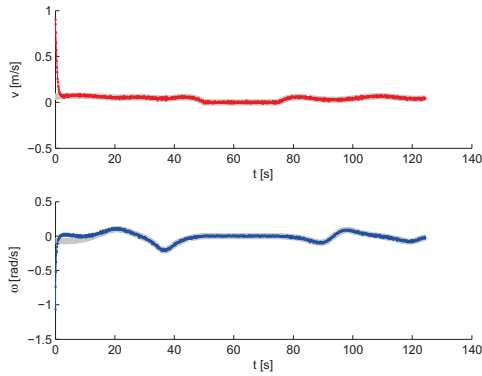
Slika 5.5: Sledenje trajektoriji s prediktivnim regulatorjem pri neničelnem začetnem pogrešku ($e_x(0) = 0,5$ m, $e_y(0) = 0,5$ m in $e_\phi(0) = 0,5$ rad). Opisi slik (a) do (d) so podani v opisu na sliki 5.4.



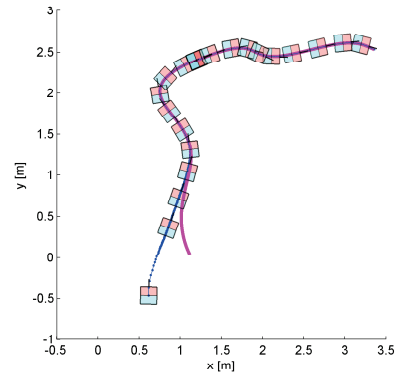
(a)



(b)

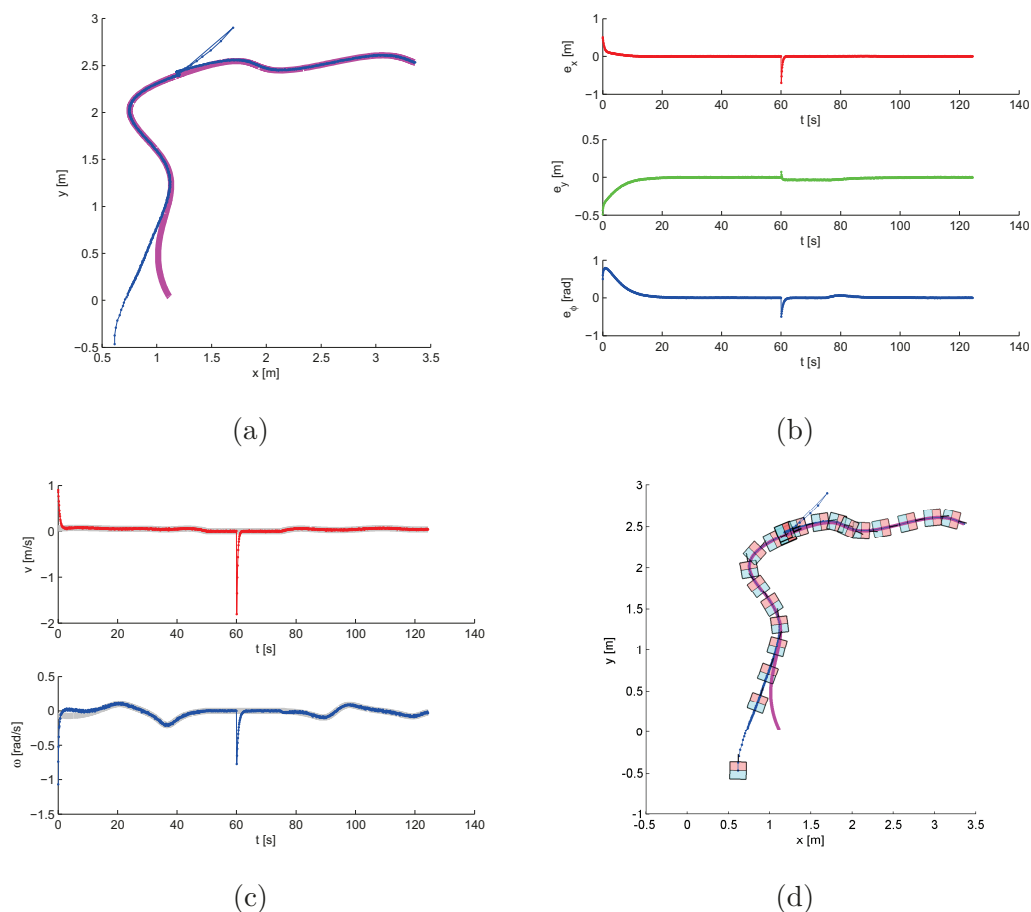


(c)



(d)

Slika 5.6: Sledenje trajektoriji s prediktivnim regulatorjem pri neničelnem začetnem pogrešku ($e_x(0) = 0,5\text{ m}$, $e_y(0) = 0,5\text{ m}$ in $e_\varphi(0) = 0,5\text{ rad}$) in normalno porazdeljenemu belemu šumu s standardno deviacijo $\sigma_{n,x} = \sigma_{n,y} = 0,002\text{ m}$ in $\sigma_{n,\varphi} = 0,002\text{ rad}$ na izhodu. Opisi slik (a) do (d) so podani v opisu na sliki 5.4.



Slika 5.7: Sledenje trajektoriji s prediktivnim regulatorjem pri neničelnem začetnem pogrešku ($e_x(0) = 0,5\text{ m}$, $e_y(0) = 0,5\text{ m}$ in $e_\phi(0) = 0,5\text{ rad}$) in normalno porazdeljenemu belemu šumu s standardno deviacijo $\sigma_{n,x} = \sigma_{n,y} = 0,002\text{ m}$ in $\sigma_{n,\phi} = 0,002\text{ rad}$ na izhodu. V trenutku $t = 60\text{ s}$ nastopi na izhodu motnja, ki izmakne mobilni sistem iz svoje lege. Opisi slik (a) do (d) so podani v opisu na sliki 5.4.

Tabela 5.1: Cenilke pri sledenju trajektoriji s prediktivnim regulatorjem za (a) primer s slike 5.4, (b) primer s slike 5.5, (c) primer s slike 5.6 in (d) primer s slike 5.7.

(a)

x	$\sqrt{e_x^2 + e_y^2}$ [m]	e_φ [rad]	v_B [m/s]	ω_B [rad/s]
$SSE(x)$	0,011 914 1	0,010 233 4	0,042 784	0,060 208 9
$SAE(x)$	3,394 15	2,835 31	5,733 81	6,616 19
$TSSE(x)$	0,739 784	0,571 164	2,659 44	3,299 29
$TSAE(x)$	210,627	166,392	356,57	385,493

(b)

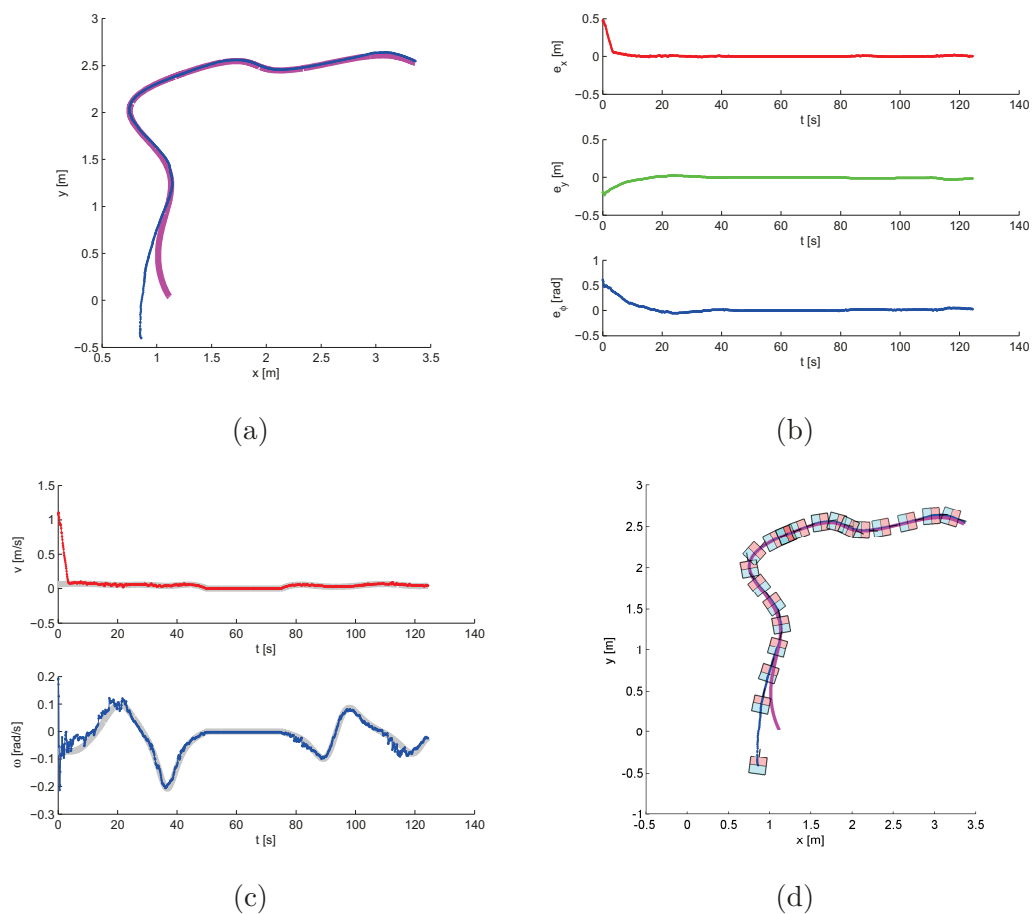
x	$\sqrt{e_x^2 + e_y^2}$ [m]	e_φ [rad]	v_B [m/s]	ω_B [rad/s]
$SSE(x)$	5,961 92	27,804	2,845 88	2,276 93
$SAE(x)$	24,1293	56,6521	5,965 27	11,2781
$TSSE(x)$	12,4405	87,7287	0,664 812	2,775 64
$TSAE(x)$	171,584	364,891	19,3518	92,0743

(c)

x	$\sqrt{e_x^2 + e_y^2}$ [m]	e_φ [rad]	v_B [m/s]	ω_B [rad/s]
$SSE(x)$	5,974 49	27,8122	2,893 62	2,337 98
$SAE(x)$	25,8771	57,6102	11,0414	15,8471
$TSSE(x)$	13,1131	88,169	3,325 04	6,036 97
$TSAE(x)$	310,364	444,565	359,77	424,534

(d)

x	$\sqrt{e_x^2 + e_y^2}$ [m]	e_φ [rad]	v_B [m/s]	ω_B [rad/s]
$SSE(x)$	7,3632	28,9265	10,2464	4,440 97
$SAE(x)$	36,5604	66,694	17,9868	21,3329
$TSSE(x)$	99,9229	160,481	446,14	133,168
$TSAE(x)$	1069,91	1135,11	779,504	767,141

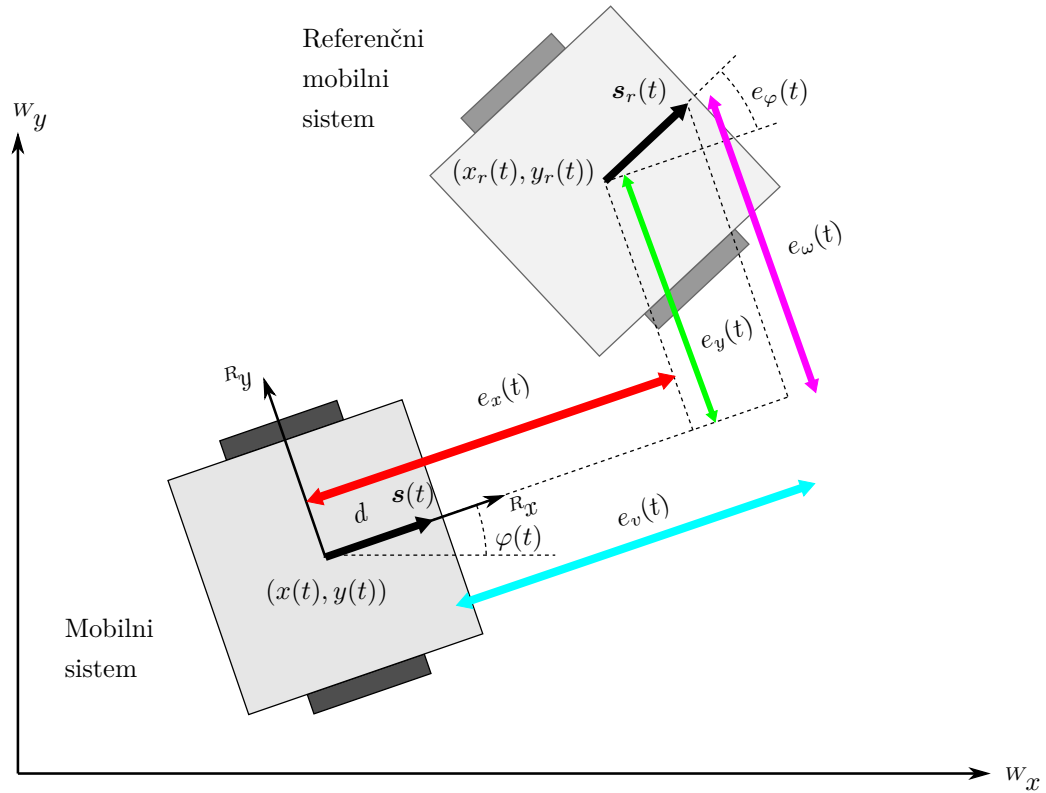


Slika 5.8: Sledenje trajektoriji s prediktivnim regulatorjem na realnem mobilnem sistemu. Opisi slik (a) do (d) so podani v opisu na sliki 5.4.

Tabela 5.2: Cenilke pri sledenju trajektoriji s prediktivnim regulatorjem na realnem mobilnem sistemu.

x	$\sqrt{e_x^2 + e_y^2}$ [m]	e_ϕ [rad]	v_B [m/s]	ω_B [rad/s]	$\sqrt{e_v^2 + e_\omega^2}$ [m]
$SSE(x)$	5,6115	13,1533	15,0032	0,557 748	4,503 87
$SAE(x)$	29,884	53,6559	24,9785	13,5459	25,2702
$TSSSE(x)$	24,0386	76,4572	15,1524	9,589 29	17,4478
$TSAE(x)$	874,618	1396,47	299,677	586,163	747,716

5.4.2 Model pomožnega pogreška



Slika 5.9: Lega referenčnega mobilnega sistema glede na mobilni sistem s pomožno točko.

V tem poglavju je prikazana izpeljava malce drugačne definicije pogreška. Pri izpeljavi si pomagamo s sliko 5.9. Dobljeni model bomo poimenovali *model pomožnega pogreška*. Na tem mestu vpeljemo pomožno referenčno trajektorijo:

$$\mathbf{s}_r(t) = \mathbf{r}_r(t) + \mathbf{R}_r(t)\mathbf{d} = \begin{bmatrix} x_r(t) \\ y_r(t) \end{bmatrix} + \begin{bmatrix} \cos \varphi_r(t) & -\sin \varphi_r(t) \\ \sin \varphi_r(t) & \cos \varphi_r(t) \end{bmatrix} \begin{bmatrix} d \\ 0 \end{bmatrix}, \quad (5.51)$$

kjer je razdalja $d \geq 0$ parameter pomožne krivulje. V primeru $d = 0$ pomožna referenčna krivulja sovпада z referenčno. Pomožno referenčno krivuljo enostavno določimo iz vsake vsaj enkrat zvezno odvedljive referenčne krivulje. Vpeljimo še

položaj pomožne točke na mobilnem sistemu:

$$\mathbf{s}(t) = \mathbf{r}(t) + \mathbf{R}(t)\mathbf{d} = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix} + \begin{bmatrix} \cos \varphi(t) & -\sin \varphi(t) \\ \sin \varphi(t) & \cos \varphi(t) \end{bmatrix} \begin{bmatrix} d \\ 0 \end{bmatrix}. \quad (5.52)$$

Regulacijski zakon zasnujemo tako, da izničimo pogrešek med pomožno točko na mobilnem sistemu $\mathbf{s}(t)$ in pomožno referenčno trajektorijo $\mathbf{s}_r(t)$:

$$\mathbf{e}(t) = \begin{bmatrix} e_v(t) & e_\omega(t) \end{bmatrix}^T = \mathbf{R}^T(t)(\mathbf{s}_r(t) - \mathbf{r}(t)) - \mathbf{d} = \mathbf{R}^T(t)(\mathbf{s}_r(t) - \mathbf{s}(t)). \quad (5.53)$$

Pri tem velja opozoriti, da smo vpeljali oznaki e_v in e_ω , ki ne označujeta pogreška tangencialne in kotne hitrosti, temveč le pomožni pogrešek. Odvod pogreška (5.53) je

$$\dot{\mathbf{e}}(t) = \dot{\mathbf{R}}^T(t)(\mathbf{s}_r(t) - \mathbf{s}(t)) + \mathbf{R}^T(t)(\dot{\mathbf{s}}_r(t) - \dot{\mathbf{s}}(t)), \quad (5.54)$$

ki ga ob upoštevanju

$$\dot{\mathbf{s}}(t) = \dot{\mathbf{r}}(t) + \dot{\mathbf{R}}(t)\mathbf{d} \quad (5.55)$$

in

$$\dot{\mathbf{s}}_r(t) = \dot{\mathbf{r}}_r(t) + \dot{\mathbf{R}}_r(t)\mathbf{d} \quad (5.56)$$

zapišemo v obliki:

$$\dot{\mathbf{e}}(t) = (\dot{\mathbf{R}}^T(t)\mathbf{R}_r(t) + \mathbf{R}^T(t)\dot{\mathbf{R}}_r(t))\mathbf{d} + \dot{\mathbf{R}}^T(t)(\mathbf{r}_r(t) - \mathbf{r}(t)) + \mathbf{R}^T(t)(\dot{\mathbf{r}}_r(t) - \dot{\mathbf{r}}(t)). \quad (5.57)$$

Ob upoštevanju kinematičnega modela mobilnega sistema (5.2) in predpostavki, da referenčna trajektorija na krši kinematičnih omejitev, lahko izpeljemo model pogreška:

$$\dot{\mathbf{e}}(t) = d(\omega_r(t) - \omega(t)) \begin{bmatrix} -\sin e_\varphi(t) \\ \cos e_\varphi(t) \end{bmatrix} + \omega(t) \begin{bmatrix} e_\omega(t) - d \sin e_\varphi(t) \\ -e_v(t) - d(1 - \cos e_\varphi) \end{bmatrix} + \begin{bmatrix} v_r(t) \cos e_\varphi(t) - v(t) \\ v_r(t) \sin e_\varphi(t) \end{bmatrix}. \quad (5.58)$$

Regulirni signal zopet sestavimo kot vsoti $v(t) = v_r(t) \cos e_\varphi + v_B(t)$ in $\omega(t) = \omega_r(t) + \omega_B(t)$ in vsoti vstavimo v izraz (5.58), kar nas privede do modela pogreška:

$$\dot{\mathbf{e}}(t) = \begin{bmatrix} (e_\omega - d \sin e_\varphi)\omega_r - v_B + e_\omega\omega_B \\ -(e_v + d(1 - \cos e_\varphi))\omega_r - (d + e_v)\omega_B + v_r \sin e_\varphi \end{bmatrix}. \quad (5.59)$$

Model pogreška (5.59) lineariziramo okoli referenčne trajektorije ($e_v = 0$, $e_\omega = 0$, $v_B = 0$ in $\omega_B = 0$), kar nas privede do *modela pomožnega pogreška*:

$$\dot{\mathbf{e}}(t) = \begin{bmatrix} 0 & \omega_r \\ -\omega_r & 0 \end{bmatrix} \mathbf{e}(t) + \begin{bmatrix} -1 & 0 \\ 0 & -d \end{bmatrix} \mathbf{u}_B(t). \quad (5.60)$$

kjer je $\mathbf{u}_B^T(t) = [v_B(t) \ \omega_B(t)]$. Model pomožnega pogreška lahko iz zveznega prostora pretvorimo v diskretni prostor na podoben način, kot smo to storili v poglavju 5.4.1. Diskretni model pogreška lahko nato neposredno uporabimo v prediktivnem regulacijskem algoritmu (poglavje 5.4). Parameter $d > 0$ predstavlja dodatni parameter v (prediktivnem) regulatorju. Vrednost parametra $d > 0$ želimo, da je primerno majhna, a dovolj velika, da regulacijski algoritem ni preveč občutljiv na merilni šum.

Oglejmo si vodljivostno matriko modela pogreška (5.60):

$$\begin{bmatrix} \mathbf{B} & \mathbf{AB} \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 & -\omega_r d \\ 0 & -d & \omega_r & 0 \end{bmatrix}. \quad (5.61)$$

Matrika (5.61) je vedno polnega ranga, če le velja $d \neq 0$. Model pogreška (5.60) je torej vodljiv. Zatorej je možno izdelati regulator, ki izniči pogrešek (5.53). Kar pomeni, da lahko primerno izvedeni regulator sledi pomožni referenčni krivulji. Tu velja zopet opozoriti, da omenjena vodljivostna analiza velja le v primeru časovno nespremenljivih matrik modela pogreška. Pri točnem sledenju pomožni referenčni trajektoriji ni nujno, da dosežemo tudi točno sledenje referenčni trajektoriji — cilj regulacije še vedno ostaja sledenje referenčni trajektoriji. Načeloma velja, da če mobilni sistem s pomožno točko sledi pomožni referenčni trajektoriji, sledi mobilni sistem tudi referenčni trajektoriji. Takšna postavitev lahko zagotovi sledenje mobilnega sistema referenčni trajektoriji, če so le izpolnjeni določeni pogoji. To je možno le v primeru, da je znan točen model med pomožno in referenčno trajektorijo in je začetna lega mobilnega sistema v primerni okolici referenčne trajektorije. Če sta tangencialna in kotna hitrost mobilnega sistema enaki nič, se z odpravo pomožnega pogreška v splošnem ne odpravi tudi dejanski pogrešek. Vendar, če smo izbrali takšno referenčno trajektorijo, ki se zvezno ustavi, lahko dosežemo relativno majhen pogrešek tudi v končni stacionarni točki, če le verno sledimo pomožni referenčni trajektoriji.

Model pomožnega pogreška (5.60) lahko uporabimo za zasnovanje prediktivnega regulatorja (poglavje 5.4). Za model pogreška (5.60) je možno izdelati regulator, ki

zagotovi stabilno delovanje povratne zanke. Regulator lahko zasnujemo s pomočjo analize Lyapunovih funkcij. Vpeljimo Lyapunovo funkcijo:

$$V(t) = \frac{1}{2}e_v^2(t) + \frac{1}{2}e_\omega^2(t), \quad (5.62)$$

ki je vedno pozitivna in enaka nič le, ko je pomožni pogrešek enak nič, $e_v = e_\omega = 0$. V odvod Lyapunove funkcije (5.62) vstavimo model pogreška:

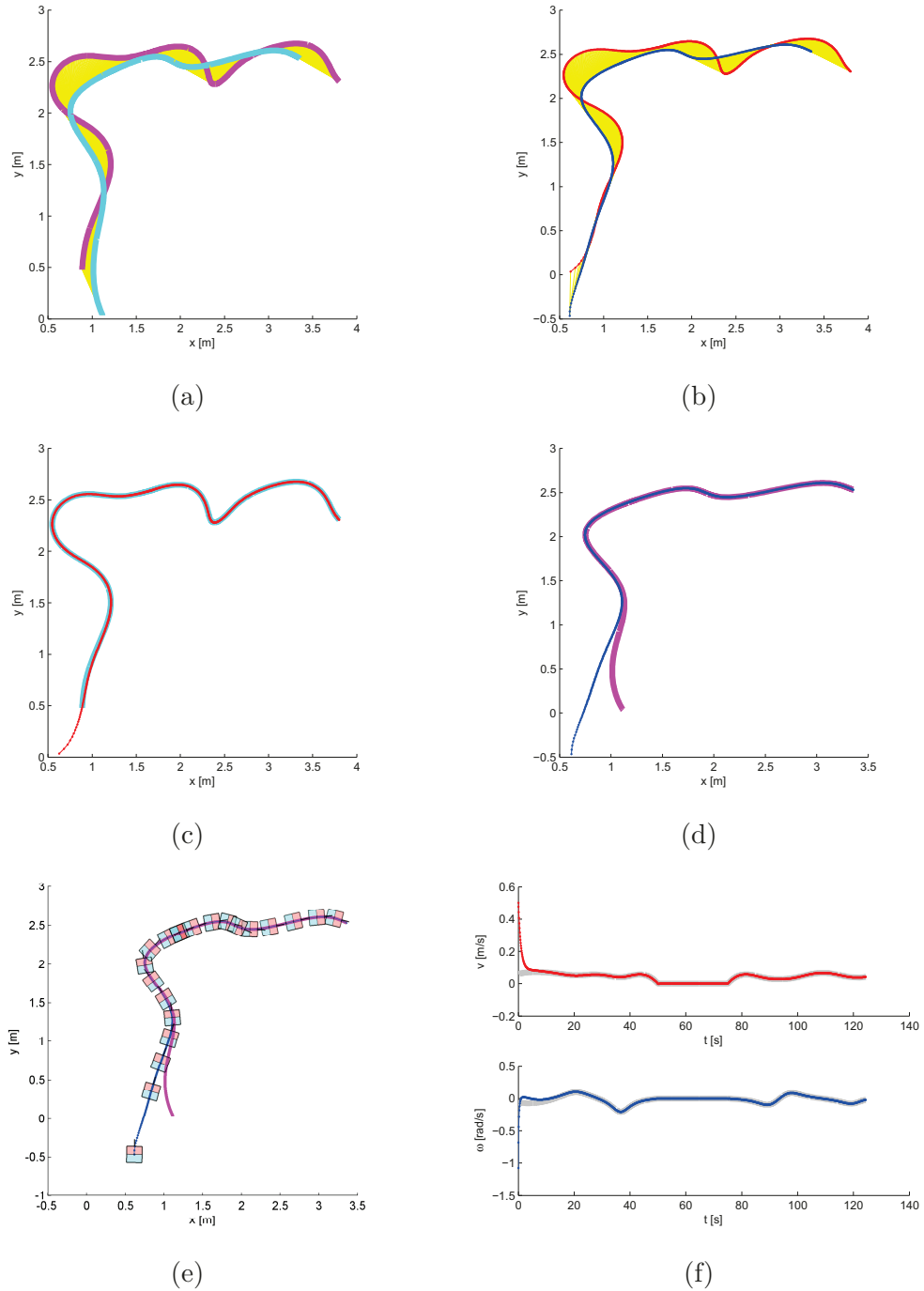
$$\dot{V}(t) = e_v(t)\dot{e}_v(t) + e_\omega(t)\dot{e}_\omega(t) = -e_v(t)v_B(t) - de_\omega(t)\omega_B(t). \quad (5.63)$$

Enostavno lahko pridemo do naslednjega regulacijskega zakona, ki pri pozitivni vrednosti konstant $k_v > 0$ in $k_\omega > 0$ zagotovi negativno semidefinitnost odvoda Lyapunove funkcije (5.63) — kar je zadosten pogoj za stabilnost zaprtozančnega sistema:

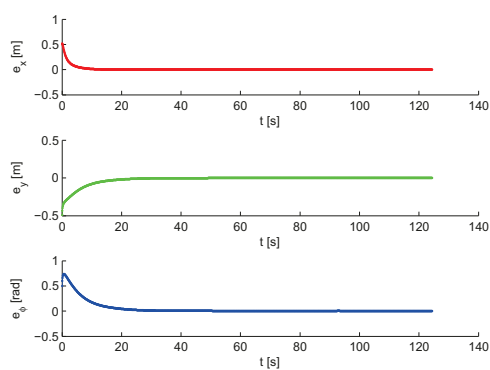
$$v_B(t) = k_v e_v(t), \quad \omega_B(t) = k_\omega \frac{e_\omega(t)}{d}. \quad (5.64)$$

Primer delovanja predstavljenega regulatorja za vrednosti parametrov $k_v = 1 \text{ s}^{-1}$, $k_\omega = 2 \text{ s}^{-1}$ in $d = 0,5 \text{ m}$ je prikazana na slikah 5.10 in 5.11.

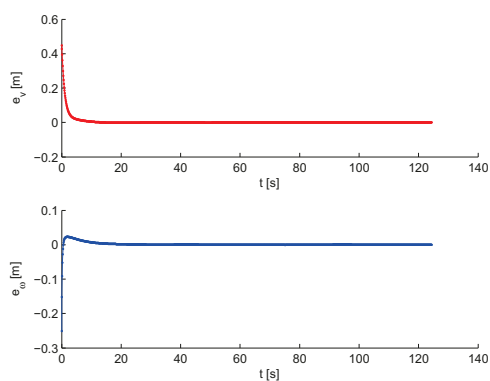
V nadaljevanju je na slikah 5.12, 5.13, 5.14 in 5.15 prikazano delovanje prediktivnega regulatorja z modelom pomožnega pogreška. Izbrani so bili sledeči parametri prediktivnega regulatorja: $h = 5$, $\mathbf{Q} = \text{diag}([5, 50]^T)$, $\mathbf{R} = \text{diag}([0,1, 0,5]^T)$, $\mathbf{A}_r = \text{diag}([0,65, 0,65]^T)$ in $d = 0,2 \text{ m}$. Delovanje regulatorja za sledenje trajektoriji je ovrednoteno še z naborom cenilk (tabela 5.3). Na sliki 5.16 je prikazano še sledenje referenčni trajektoriji s prediktivnim regulatorjem, ki je bilo izvedeno na realnem mobilnem sistemu Pioneer 3-AT. Za parametre regulatorja so bile izbrane enake vrednosti kot v simulacijskih primerih na slikah 5.12 do 5.15. Regulirni hitrosti sta bili v tem primeru omejeni: $|v| < 0,2 \text{ m/s}$ in $|\omega| < 1 \text{ rad/s}$. Delovanje regulatorja je ovrednoteno v tabeli 5.4.



Slika 5.10: Sledenje trajektoriji s pomožnim regulatorjem (5.64) pri neničelnem začetnem pogrešku ($e_x(0) = 0,5$ m, $e_y(0) = 0,5$ m in $e_\varphi(0) = 0,5$ rad). (a) Referenčna in pomožna referenčna trajektorija, (b) sledenje pomožne točke in mobilnega sistema, (c) sledenje pomožne točke pomožni referenčni trajektoriji, (d) sledenje mobilnega sistema referenčni trajektoriji, (e) lege mobilnega sistema ob vsaki polni sekundi in (f) regulirna signala.

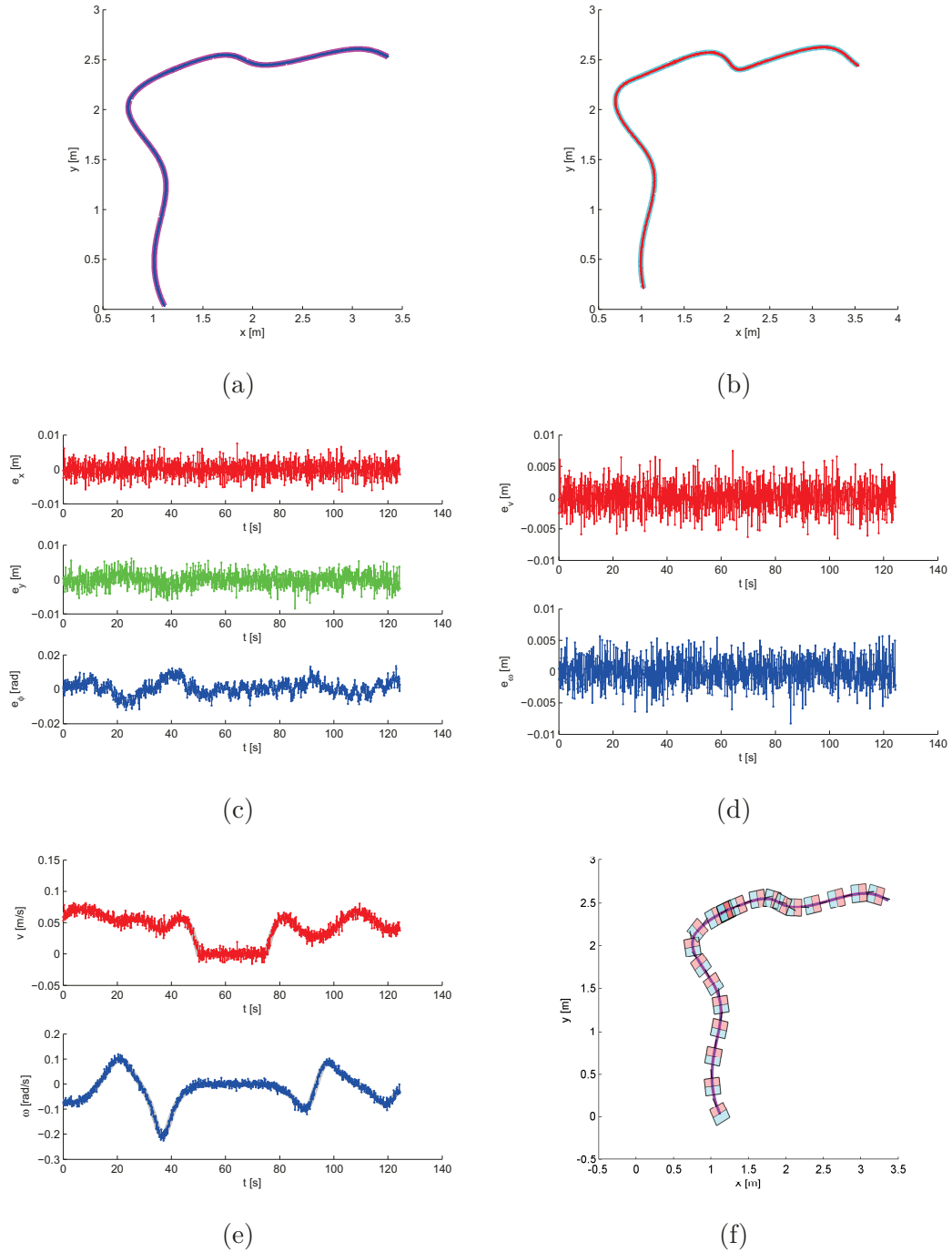


(a)

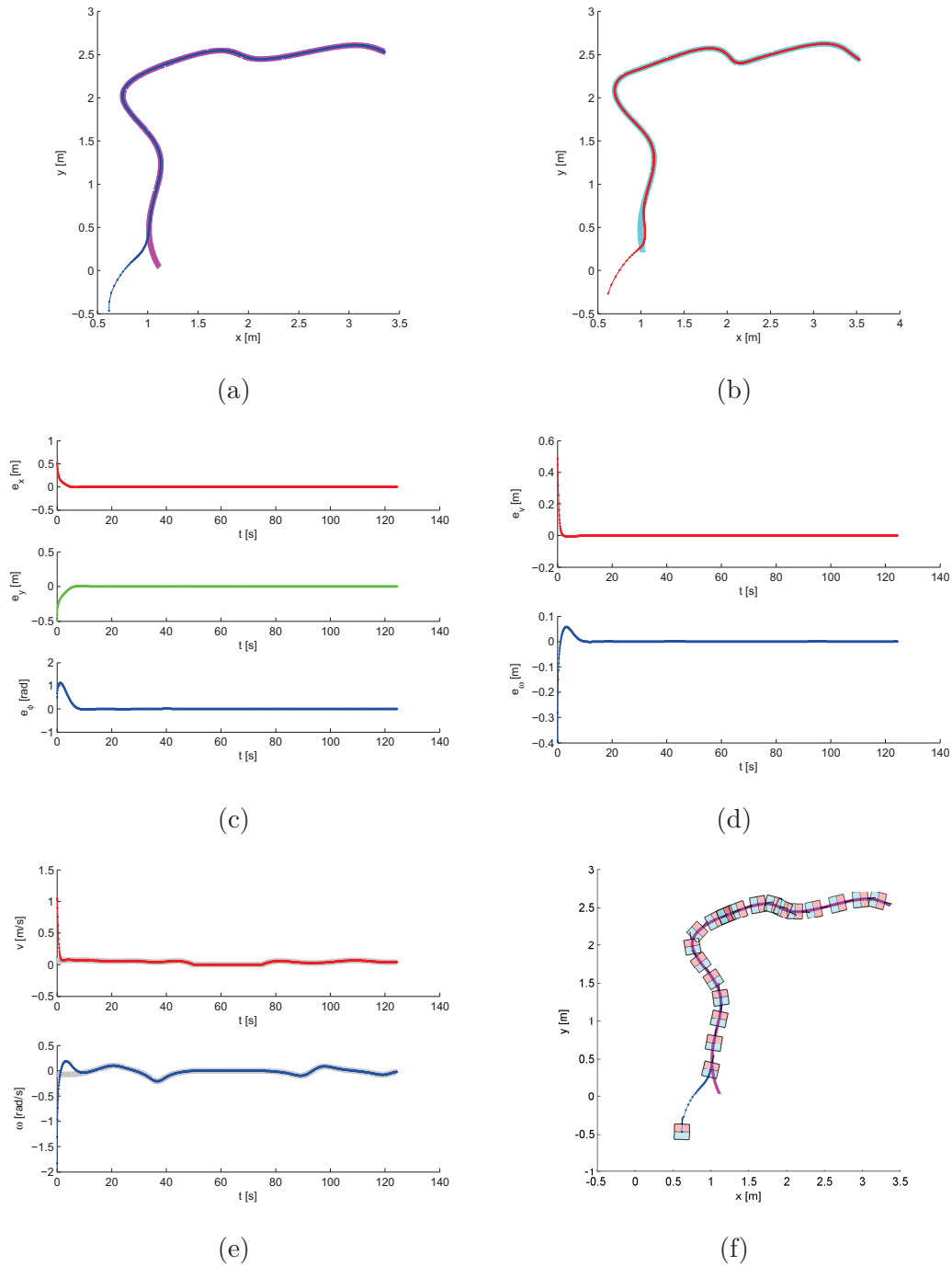


(b)

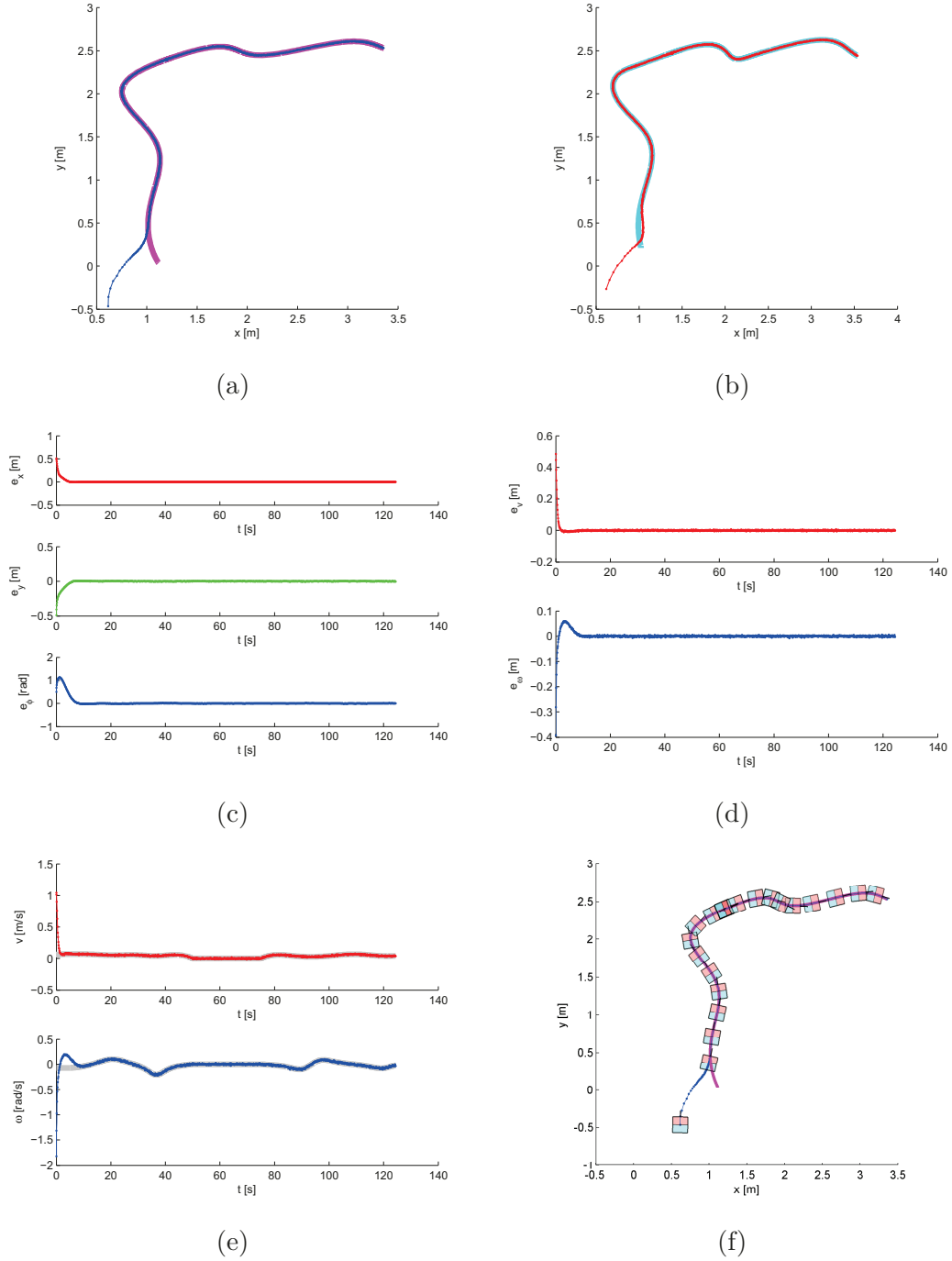
Slika 5.11: Pogrešek pri sledenju trajektoriji s pomožnim regulatorjem (5.64). (a) Časovni potek pogreška mobilnega sistema in (b) časovni potek pomožnega pogreška.



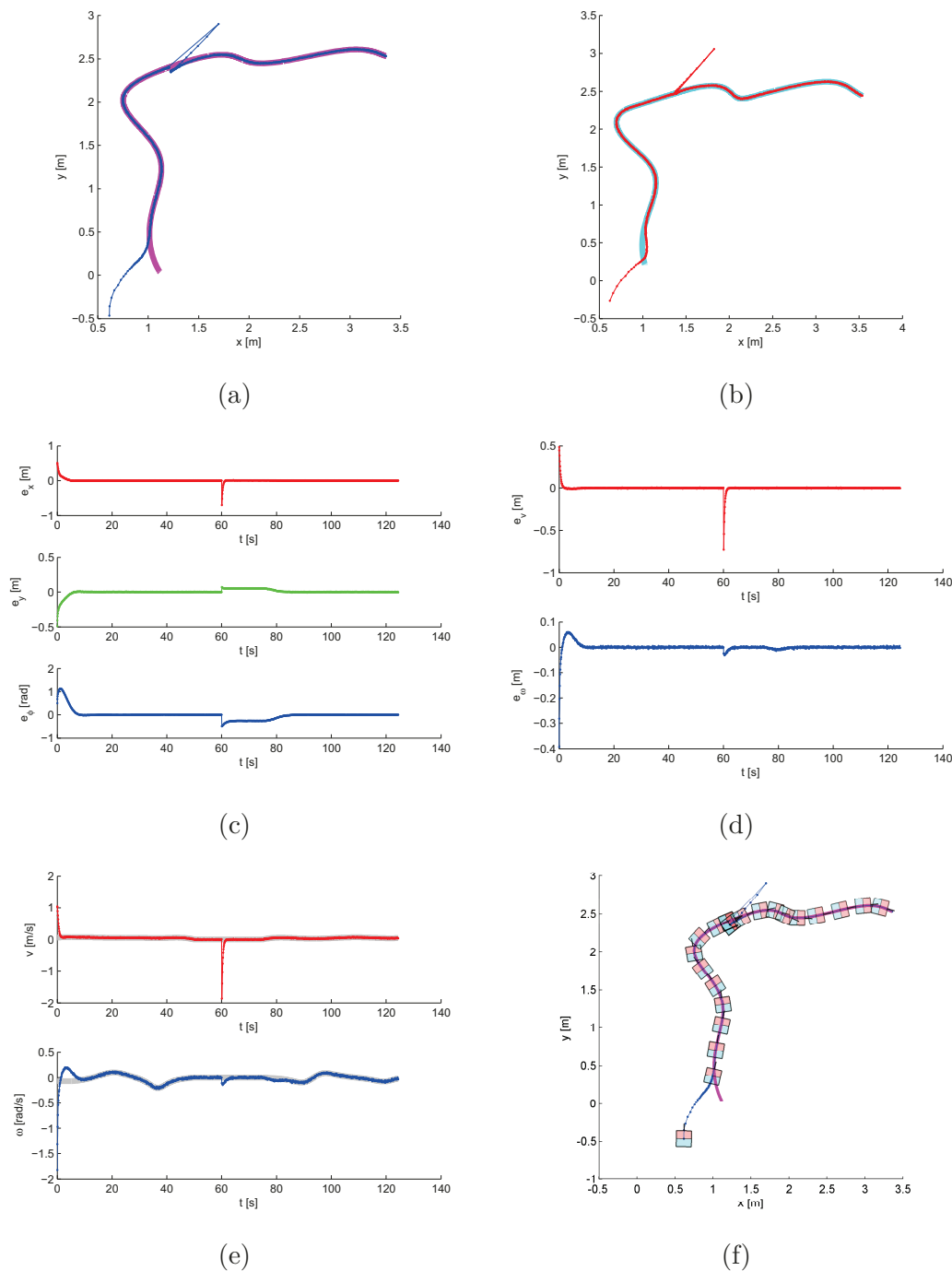
Slika 5.12: Sledenje trajektoriji s prediktivnim regulatorjem z modelom pomožnega pogreška pri ničelnem začetnem pogrešku in normalno porazdeljenemu belemu šumu s standardno deviacijo $\sigma_{n,x} = \sigma_{n,y} = 0,002$ m in $\sigma_{n,\varphi} = 0,002$ rad na izhodu. (a) Referenčna (krepko) in dejanska trajektorija, (a) referenčna pomožna (krepko) in dejanska pomožna trajektorija, (c) pogrešek, (c) pomožni pogrešek, (e) regulirna signala in (f) lege mobilnega sistema vsakih polnih pet sekund.



Slika 5.13: Sledenje trajektoriji s prediktivnim regulatorjem pri neničelnem začetnem pogrešku ($e_x(0) = 0,5$ m in $e_y(0) = 0,5$ m in $e_\phi(0) = 0,5$ rad). Opisi slik (a) do (f) so podani v opisu na sliki 5.12.



Slika 5.14: Sledenje trajektoriji s prediktivnim regulatorjem pri neničelnem začetnem pogrešku ($e_x(0) = 0,5$ m, $e_y(0) = 0,5$ m in $e_\phi(0) = 0,5$ rad) in normalno porazdeljenemu belemu šumu s standardno deviacijo $\sigma_{n,x} = \sigma_{n,y} = 0,002$ m in $\sigma_{n,\phi} = 0,002$ rad na izhodu. Opisi slik (a) do (f) so podani v opisu na sliki 5.12.



Slika 5.15: Sledenje trajektoriji s prediktivnim regulatorjem pri neničelnem začetnem pogrešku ($e_x(0) = 0,5$ m, $e_y(0) = 0,5$ m in $e_\phi(0) = 0,5$ rad) in normalno porazdeljenemu belemu šumu s standardno deviacijo $\sigma_{n,x} = \sigma_{n,y} = 0,002$ m in $\sigma_{n,\phi} = 0,002$ rad na izhodu. V trenutku $t = 60$ s nastopi na izhodu motnja, ki izmakne mobilni sistem iz svoje lege. Opisi slik (a) do (f) so podani v opisu na sliki 5.12.

Tabela 5.3: Cenilke pri sledenju trajektoriji s prediktivnim regulatorjem z modelom pomožnega pogreška za (a) primer s slike 5.12, (b) primer s slike 5.13, (c) primer s slike 5.14 in (d) primer s slike 5.15.

(a)

x	$\sqrt{e_x^2 + e_y^2}$ [m]	e_φ [rad]	v_B [m/s]	ω_B [rad/s]	$\sqrt{e_v^2 + e_\omega^2}$ [m]
$SSE(x)$	0,011 384 7	0,022 833 2	0,040 515 5	0,115 607	0,011 496 5
$SAE(x)$	3,323 25	4,180 95	5,598 06	9,6069	3,357 44
$TSSE(x)$	0,707 504	1,284 35	2,540 46	7,159 62	0,717 871
$TSAE(x)$	206,206	247,086	350,076	593,27	208,59

(b)

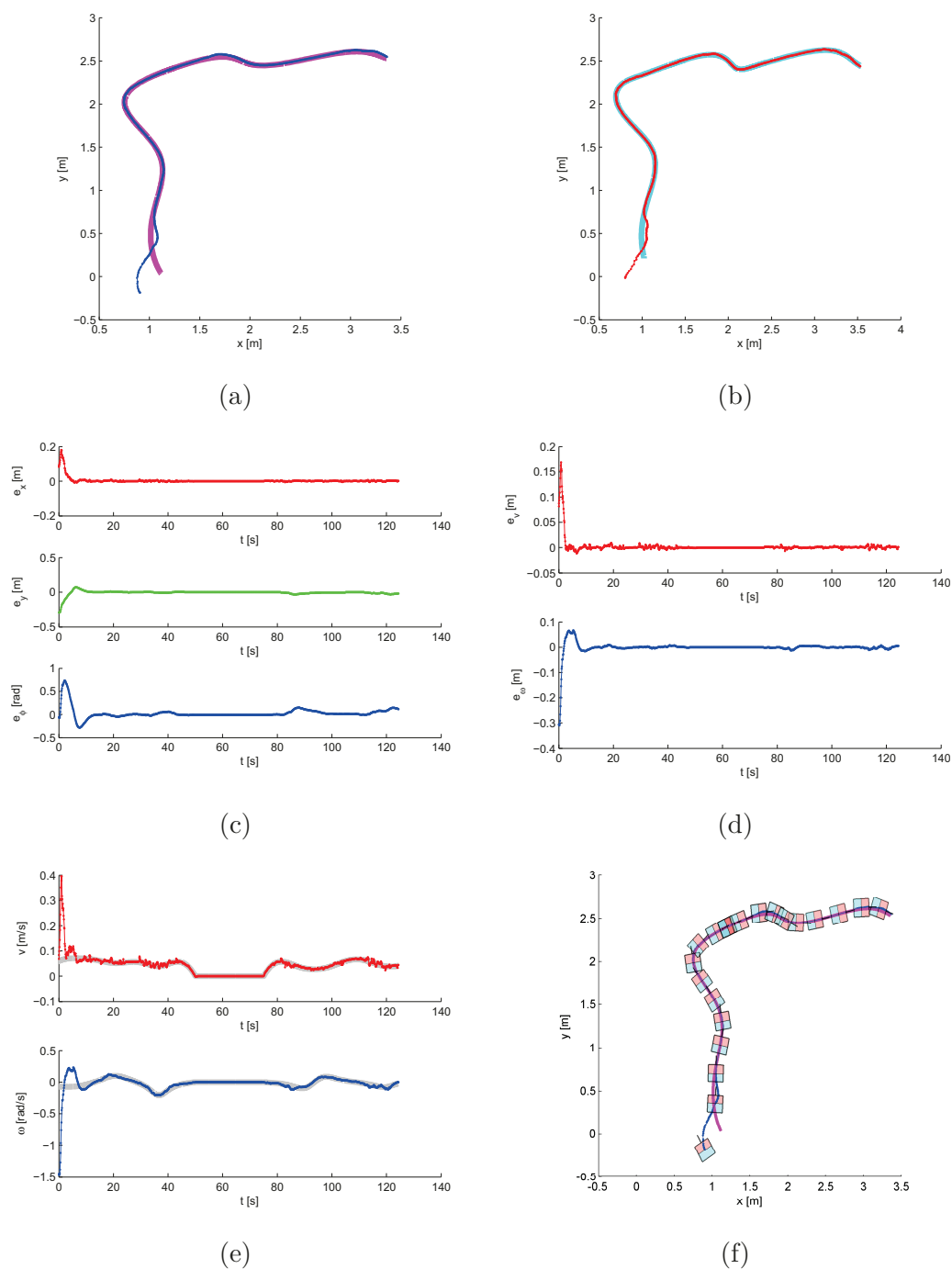
x	$\sqrt{e_x^2 + e_y^2}$ [m]	e_φ [rad]	v_B [m/s]	ω_B [rad/s]	$\sqrt{e_v^2 + e_\omega^2}$ [m]
$SSE(x)$	2,969 39	38,4056	3,851 66	8,621 47	1,294 54
$SAE(x)$	10,2337	49,0706	6,185 59	18,7375	5,936 93
$TSSE(x)$	2,109 27	77,1659	0,806 691	9,415 68	0,647 841
$TSAE(x)$	48,1507	284,444	13,3926	106,218	25,9589

(c)

x	$\sqrt{e_x^2 + e_y^2}$ [m]	e_φ [rad]	v_B [m/s]	ω_B [rad/s]	$\sqrt{e_v^2 + e_\omega^2}$ [m]
$SSE(x)$	2,982 81	38,4399	3,8985	8,742 41	1,307 21
$SAE(x)$	12,7535	50,211	11,4048	26,5515	8,802 11
$TSSE(x)$	2,799 57	77,9253	3,350 45	16,5405	1,364 32
$TSAE(x)$	220,441	365,061	352,439	639,27	219,362

(d)

x	$\sqrt{e_x^2 + e_y^2}$ [m]	e_φ [rad]	v_B [m/s]	ω_B [rad/s]	$\sqrt{e_v^2 + e_\omega^2}$ [m]
$SSE(x)$	4,550 17	53,5819	11,6762	9,027 35	2,5018
$SAE(x)$	24,3281	106,058	18,5898	30,3818	12,0862
$TSSE(x)$	101,557	1114,12	471,752	35,4036	73,3898
$TSAE(x)$	1006,94	4262,31	786,711	908,211	425,051



Slika 5.16: Sledenje trajektoriji s prediktivnim regulatorjem z modelom pomožnega pogreška na realnem mobilnem sistemu. Opisi slik (a) do (f) so podani v opisu na sliki 5.12.

Tabela 5.4: Cenilke pri sledenju trajektoriji s prediktivnim regulatorjem z modelom pomožnega pogreška na realnem mobilnem sistemu.

x	$\sqrt{e_x^2 + e_y^2}$ [m]	e_φ [rad]	v_B [m/s]	ω_B [rad/s]	$\sqrt{e_v^2 + e_\omega^2}$ [m]
$SSE(x)$	1,826 19	17,7553	0,960 898	18,8589	1,215 99
$SAE(x)$	16,4874	66,585	8,655 85	38,9302	10,3168
$TSSE(x)$	12,563	285,238	2,720 46	37,8017	2,329 42
$TSAE(x)$	614,928	2909,23	231,455	912,607	237,865

6 Načrtovanje poti

6.1 Parametrične krivulje

Za načrtovanje poti se pogosto uporabljajo parametrične krivulje. Običajno gre za zvezne krivulje z zveznimi odvodi do vsaj določenega reda, kar omogoča analitično odvajanje krivulje po parametru. Omenili smo že, da za neholonomične mobilne sisteme velja lastnost t. i. diferencialne ploskosti, kar pomeni, da lahko vse vhode in stanja sistema zapišemo s končnim številom odvodov izhodov sistema. Izhod mobilnega sistema je njegov položaj v prostoru, kar dejansko pomeni, da lahko izrazimo vhode v sistem (odprtozančne regulirne signale), če imamo na voljo krivuljo in določeno število njenih odvodov (običajno zadostujeta že prvi in drugi odvod). Parametrične krivulje se običajno da združiti v zlepke, kar omogoča skoraj poljubno definicijo poti. V nadaljevanju je podrobneje predstavljena posebna vrsta parametričnih krivulj — Bernstein-Bézierjeve krivulje —, ki imajo nekaj zanimivih lastnosti, ki jih lahko s pridom uporabimo v postopku načrtovanja poti.

Splošna parametrična krivulje $\mathbf{r}(s) \in \mathbb{R}^n$ je običajno definirana na omejenem intervalu $s \in [s_a, s_b]$, kjer je parameter s praviloma monotonno naraščajoča funkcija, ki je odvisna od časa, $s = s(t)$. Če velja $s_a = s(t_a)$ in $s_b = s(t_b)$, je krivulja definirana na časovnem intervalu $t \in [t_a, t_b]$. Odvod splošne parametrične krivulje lahko določimo s pravilom posrednega odvajanja:

$$\frac{d\mathbf{r}(s(t))}{dt} = \frac{d\mathbf{r}(s)}{ds} \frac{ds(t)}{dt}. \quad (6.1)$$

Bistvo enačbe (6.1) je, da lahko obliko krivulje in njen časovni potek načrtujemo ločeno. Kar ima za posledico, da za določeno krivuljo obstaja cela družina različnih časovnih funkcij, ki dejansko opisujejo hitrost gibanja in pospešek po krivulji.

Brez izgube na splošnosti lahko uvedemo normaliziran čas $\lambda = \frac{t-t_a}{\Lambda}$, kjer je $\Lambda = t_b - t_a > 0$ čas *trajanja* trajektorije — velja $t_a \leq t \leq t_b$ oz. $t_a \leq t \leq t_a + \Lambda$.

Torej zavzame λ vrednosti med 0 in 1. V tem primeru je odvod krivulje (6.1) sledeč:

$$\frac{d\mathbf{r}(s(\lambda(t)))}{dt} = \frac{1}{\Lambda} \frac{d\mathbf{r}(s)}{ds} \frac{ds}{d\lambda}. \quad (6.2)$$

Če poiščemo še preostale višje odvode, enostavno ugotovimo, da ima splošni odvod reda $d \geq 0$ obliko:

$$\frac{d^d \mathbf{r}(s(\lambda(t)))}{dt^d} = \frac{1}{\Lambda^d} \frac{d^d \mathbf{r}(\lambda)}{d\lambda^d}. \quad (6.3)$$

Vrednost rezultata (6.3) je v tem, da lahko zmanjšamo vrednost odvodov krivulje pod poljubno mejo, tako da ustrezno povečamo čas trajanja krivulje Λ . Na podlagi ločitvenega teorema (6.1) lahko torej načrtujemo obliko krivulje neodvisno od njenega časovnega poteka, časovni potek pa lahko prilagodimo tako, da zadostimo določenim omejitvam, kot je prikazano v naslednjem poglavju.

6.1.1 Optimizacija trajektorije

Ko je pot $\mathbf{r}(s)$, $s \in [s_a, s_b]$ določena, moramo določiti še časovno funkcijo $s = s(t)$ ($s(t_a) = s_a$, $s(t_b) = s_b$, $t_a \leq t \leq t_b$), ki določa kako bo mobilni sistem poti sledil [44]. Pri izbiri časovne funkcije želimo običajno ustreči določenim omejitvam, kot so npr. maksimalna tangencialna hitrost v_{max} , maksimalni tangencialni pospešek a_{max} , maksimalna kotna hitrost ω_{max} , maksimalni kotni pospešek α_{max} itd. Pogoji, ki zadosti vsem pravkar omenjenim omejitvam, zapišemo takole:

$$|v(t)| < v_{max} \wedge |a(t)| < a_{max} \wedge |\omega(t)| < \omega_{max} \wedge |\alpha(t)| < \alpha_{max} \quad \forall t. \quad (6.4)$$

Če upoštevamo normaliziran čas λ lahko tangencialno in kotno hitrost zapišemo kot:

$$v(t) = v_s(s) \frac{ds}{d\lambda} \frac{d\lambda}{dt} = v_s(s) \frac{ds}{d\lambda} \frac{1}{\Lambda}, \quad (6.5)$$

$$\omega(t) = \omega_s(s) \frac{ds}{d\lambda} \frac{d\lambda}{dt} = \omega_s(s) \frac{ds}{d\lambda} \frac{1}{\Lambda}. \quad (6.6)$$

Podobno sledi tudi za tangencialni in kotni pospešek:

$$a(t) = \left(a_s(s) \left(\frac{ds}{d\lambda} \right)^2 + v_s(s) \frac{d^2 s}{d\lambda^2} \right) \frac{1}{\Lambda^2}, \quad (6.7)$$

$$\alpha(t) = \left(\alpha_s(s) \left(\frac{ds}{d\lambda} \right)^2 + \omega_s(s) \frac{d^2 s}{d\lambda^2} \right) \frac{1}{\Lambda^2}. \quad (6.8)$$

Na podlagi rezultatov (6.5), (6.6), (6.7) in (6.8) lahko zaključimo, da je možno z ustreznim skaliranjem časa t , zadostiti vsem zahtevam (6.4), saj se z večanjem faktorja Λ manjšajo absolutne vrednosti hitrosti in pospeškov. Pri tem se časovne funkcije hitrosti in pospeškov v vseh časovnih trenutkih skalirajo z enakim faktorjem: tangencialna in kotna hitrost s faktorjem $\frac{1}{\Lambda}$, tangencialni in kotni pospešek pa s faktorjem $\frac{1}{\Lambda^2}$. Pri tem velja ponovno poudariti, da takšen način skaliranja ne vpliva na obliko poti. Praktično nas običajno zanima najkrajši čas, pri katerem zadostimo vsem zahtevam (6.4). Torej smo prispeli do optimizacijskega problema, kjer iščemo minimalni čas Λ , ki zadosti vseh zahtevam (6.4). Izkaže se, da je postopek optimizacije še posebej preprost, saj je optimalna vrednost časa Λ podana kar z izrazom:

$$\Lambda_{opt.} = \max \left(\left\{ \frac{\max_s |v_s(s)|}{v_{max}}, \frac{\max_s |\omega_s(s)|}{\omega_{max}}, \sqrt{\frac{\max_s |a_s(s)|}{a_{max}}}, \sqrt{\frac{\max_s |\alpha_s(s)|}{\alpha_{max}}} \right\} \right) \quad (6.9)$$

Torej s katerokoli izbiro $\Lambda \geq \Lambda_{opt.}$ zadostimo omejitvam (6.4). Če za določeno veličino v (6.4) ne zahtevamo omejitve, lahko iz izraza (6.9) enostavno odstranimo element, ki pripada tej veličini. Funkcije vseh veličin, ki nastopajo v enačbi (6.9), so zvezne in odvedljive na intervalu $s \in [s_a, s_b]$. Funkcije lahko zavzamejo maksimalno vrednost bodisi v ekstremnih točkah bodisi v robnih točkah ($s \in \{s_a, s_b\}$) — pri tem nas zanimajo le točke znotraj intervala $s \in [s_a, s_b]$. To pomeni, da moramo za izračun maksimalne vrednosti posamezne funkcije preveriti končno število vrednosti funkcije.

V primeru, ko za časovno funkcijo izberemo preprosto linearno zvezo $s(\lambda) = \lambda$, zavzamejo enačbe (6.5), (6.6), (6.7) in (6.8) precej enostavno obliko:

$$v(t) = \frac{1}{\Lambda} v_\lambda(\lambda), \quad \omega(t) = \frac{1}{\Lambda} \omega_\lambda(\lambda), \quad (6.10)$$

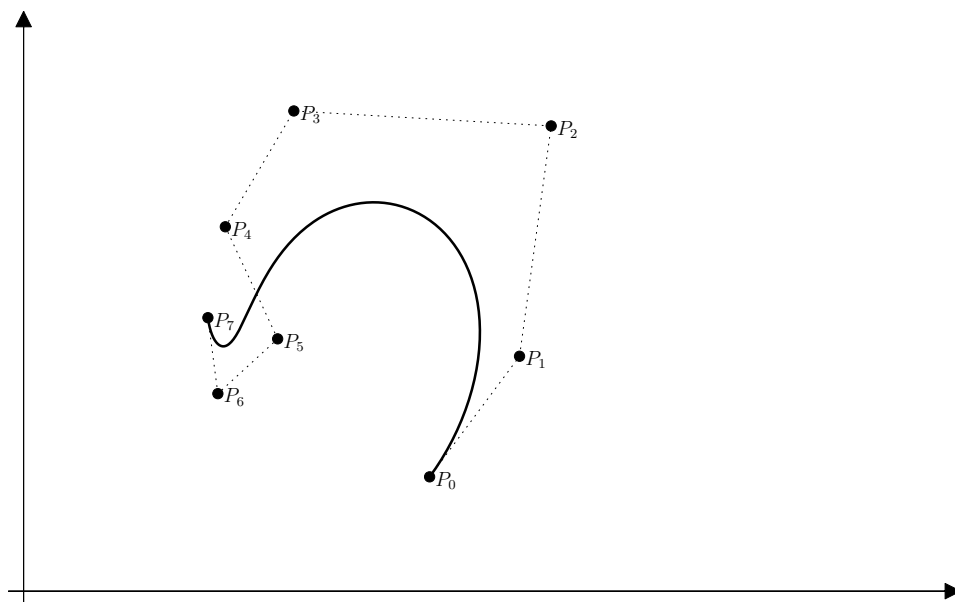
$$a(t) = \frac{1}{\Lambda^2} a_\lambda(\lambda), \quad \alpha(t) = \frac{1}{\Lambda^2} \alpha_\lambda(\lambda). \quad (6.11)$$

6.2 Bernstein-Bézierjeve krivulje

Zapišimo splošno Bernstein-Bézierjevo (BB) krivuljo reda $b \in \mathbb{N}$ v parametrični obliki:

$$\mathbf{r}_\lambda(\lambda) = \sum_{i=0}^b B_{i,b}(\lambda) \mathbf{p}_i, \quad B_{i,b}(\lambda) = \binom{b}{i} \lambda^i (1-\lambda)^{b-i}, \quad (6.12)$$

kjer parameter $\lambda \in [0, 1]$ predstavlja normaliziran čas, ki je preko linearne relacije $t = t_a + \lambda\Lambda$ povezan z absolutnim časom t . Parameter Λ predstavlja trajanje krivulje od začetka ($\lambda = 0$) do konca ($\lambda = 1$) krivulje. Funkcija $B_{i,b}(\lambda)$ je t. i. Bernsteinov polinom za katerega velja $\sum_{i=0}^b B_{i,b}(\lambda) = 1$ za vsak $\lambda \in [0, 1]$. Krivulja (6.12) spada v razred gladkih krivulj $\mathcal{C}^\infty$ in je popolnoma definirana z množico $b + 1$ kontrolnih točk $\{\mathbf{p}_i\}_{i=0,1,\dots,b}$. Bézierjeva krivulja se vedno nahaja znotraj konveksne ovojnice (t. i. Bézierjevega poligona), ki je napeta preko vseh kontrolnih točk.



Slika 6.1: Primer BB-krivulje s kontrolnimi točkami.

Njen odvod reda $d \geq 0$ glede na absolutni čas t je definiran, kot sledi:

$$\frac{d^d \mathbf{r}(t)}{dt^d} = \frac{1}{\Lambda^d} \frac{d^d \mathbf{r}_\lambda(\lambda)}{d\lambda^d}. \quad (6.13)$$

Odvod BB-krivulje je zopet BB-krivulja:

$$\frac{d\mathbf{r}_\lambda(\lambda)}{d\lambda} = b \sum_{i=0}^{b-1} B_{i,b-1}(\lambda) (\mathbf{p}_{i+1} - \mathbf{p}_i). \quad (6.14)$$

Splošen odvode krivulje (6.12) gleda na normaliziran čas λ ($0 < \lambda < 1$) lahko zapišemo na sledeč način:

$$\frac{d^d \mathbf{r}_\lambda(\lambda)}{d\lambda^d} = d! \sum_{i=0}^b \sum_{j=0}^d \left\{ \begin{matrix} i \\ d-j \end{matrix} \right\} \left\{ \begin{matrix} b-i \\ j \end{matrix} \right\} \lambda^{-d+j} (\lambda-1)^{-j} B_{i,b}(\lambda) \mathbf{p}_i, \quad (6.15)$$

kjer smo uporabili posebno razširitev binomskega koeficienta:

$$\left\{ \begin{matrix} y \\ x \end{matrix} \right\} = \begin{cases} \binom{y}{x}; & y \geq x \wedge x \geq 0, \\ 0; & \text{šicer,} \end{cases} \quad \forall \quad x, y \in \mathbb{Z}. \quad (6.16)$$

Pri tem moramo opozoriti, da je potrebno posebno pozornost nameniti primeroma, ko se vrednost λ nahaja blizu začetne oz. končne točke, saj na teh kritičnih mestih nastopi singularnost (določeni členi v enačbi (6.15) zavzamejo obliko 0^0), zato ne moremo neposredno uporabiti enačbe (6.15). V robnih točkah lahko izpeljemo naslednji relaciji ($d \geq 0, b \geq d$):

$$\lim_{\lambda \rightarrow 0^+} \frac{d^d \mathbf{r}_\lambda(\lambda)}{d\lambda^d} = \binom{b}{d} d! \sum_{i=0}^d (-1)^i \binom{d}{i} \mathbf{p}_{d-i}, \quad (6.17)$$

$$\lim_{\lambda \rightarrow 1^-} \frac{d^d \mathbf{r}_\lambda(\lambda)}{d\lambda^d} = \binom{b}{d} d! \sum_{i=0}^d (-1)^i \binom{d}{i} \mathbf{p}_{b-i}. \quad (6.18)$$

Zaključimo lahko, da je odvod reda $d \geq 0$ v robnih točkah odvisen le od prvih oz. zadnjih $d + 1$ kontrolnih točk. Iz (6.17) in (6.18) lahko tudi zaključimo, da lahko povsem enako obravnavamo lastnosti BB-krivulje v začetni in končni robni točki.

Za BB-krivuljo velja teorem robne interpolacije, kar pomeni, da se BB-krivulja vedno začne ($\lambda = 0$) v začetni kontrolni točki $\mathbf{p}_0$ in konča ($\lambda = 1$) v končni kontrolni točki. Vse ostale vmesne kontrolne točke pa v splošnem ne ležijo na krivulji. Ker sta robni točki BB-krivulje hkrati tudi kontrolni točki le-te, lahko z zlepki dosežemo, da izdelamo krivuljo, ki poteka skozi določeno množico točk. Poljubno BB-krivuljo lahko v poljubni točki razdelimo v dve BB-krivulji, pri čimer nam je v pomoč De Casteljauev algoritem [170, 171]. Med drugim velja še, da lahko vsako BB-krivuljo reda b predstavimo z BB-krivuljo višjega reda $b + 1$, vendar pa obratni postopek ni mogoč, torej BB-krivulje reda b ne moremo predstaviti z BB-krivuljo reda nižjega reda $b - 1$.

6.2.1 Posebne lastnosti v 2D

V tem razdelku so obravnavane nekatere posebne lastnosti BB-krivulj v 2D-prostoru. Zanimata nas predvsem tangencialna in kotna hitrost po krivulji ter nekateri njuni višji odvodi oz. integrali.

V 2D-primeru ima točka na krivulji le dve komponenti: $\mathbf{r}(t) = [x(t) \ y(t)]^T$. Tangencialna hitrost v pa je definirana, kot sledi:

$$v(t) = \sqrt{\dot{x}^2(t) + \dot{y}^2(t)}. \quad (6.19)$$

Glede na definicijo (6.19) je tangencialna hitrost vedno pozitivna: $v(t) \geq 0 \ \forall \ t$. Če predpostavimo, da je tangencialna hitrost striktno večja od nič ($v(t) > 0$) za vse čase t , potem lahko definiramo smer krivulje φ :

$$\varphi(t) = \arctan \frac{\dot{y}(t)}{\dot{x}(t)} + C\pi, \quad C \in \mathbb{Z}, \quad (6.20)$$

kotno hitrost ω :

$$\omega(t) = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{\dot{x}^2(t) + \dot{y}^2(t)} = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{v^2(t)}. \quad (6.21)$$

in tangencialni pospešek a :

$$a(t) = \frac{\dot{x}(t)\ddot{x}(t) + \dot{y}(t)\ddot{y}(t)}{\sqrt{\dot{x}^2(t) + \dot{y}^2(t)}} = \frac{\dot{x}(t)\ddot{x}(t) + \dot{y}(t)\ddot{y}(t)}{v(t)}. \quad (6.22)$$

Utegne nas zanimati tudi ukrivljenost krivulje κ , ki je definirana z razmerjem:

$$\kappa(t) = \frac{\omega(t)}{v(t)} = \frac{\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t)}{(\dot{x}^2(t) + \dot{y}^2(t))^{\frac{3}{2}}}. \quad (6.23)$$

Vrednosti (6.19), (6.20), (6.21) in (6.23) lahko izrazimo s kontrolnimi točkami, če v slednje izraze vstavimo primerne odvode (6.13). Pri tem moramo posebno pozornost nameniti robnim točkam, kar je podrobneje opisano v naslednjih podpodglavjih. V primeru, da je tangencialna hitrost vedno striktno pozitivna ($v(t) > 0$) — torej je pozitivna tudi v robnih točkah —, v robnih točkah ne pride do singularnosti, zato tak primer imenonujemo *robna točka ničte vrste*. Izkaže se, da je v robnih točkah možno obravnavati lastnosti krivulje tudi, ko je tangencialna hitrost v robni točki enaka nič — torej bodisi $\lim_{t \rightarrow t_a^+} v(t) = 0$ bodisi $\lim_{t \rightarrow t_b^-} v(t) = 0$. Take singularne primere označimo z izrazom *robne točke višje vrste*. Vrsto robnih točk določimo glede na prvo neničelno tangencialno hitrost v začetni točki. Obravnava robnih točk višjega reda je pomembna predvsem zato, ker omogoča konstruiranje krivulje s poljubnim začetnim in poljubnim končnim robnim pogojem. Torej lahko načrtamo tudi krivulje, ki imajo določene odvode v robnih točkah enake nič.

Pokažemo lahko, da je tangencialna hitrost poljubne BB-krivulje lahko enaka nič le v robnih točkah, sicer pa je vedno striktno večja od nič, če le niso vse kontrolne točke zbrane v isti točki — to je tudi edini posebni primer, ko je tangencialna hitrost enaka nič po celi v točko izrojeni krivulji. Izrek je pomemben s stališča, da regulatorji za vodenje po trajektoriji običajno zahtevajo neničelno tangencialno hitrost. Tangencialna hitrost se lahko približa ničelni vrednosti poljubno blizu, vendar je nikoli ne doseže. V postopku načrtovanja poti zato običajno želimo, da je hitrost po trajektoriji povsod večja od minimalne predpisane vrednosti, kar običajno zagotovimo s postopkom optimizacije poti.

Pri obravnavi nas zanimajo mejne hitrosti, ki jih dosežemo na krivulji, saj običajno želimo doseči gibanje v predpisanih mejnih hitrostih. Hitrostni profil krivulje lahko spreminjamo s premikanjem kontrolnih točk, kar ima vpliv tudi na obliko krivulje. Če želimo spremeniti le hitrostni profil krivulje, lahko to dosežemo s spreminjanjem parametra Λ , ki določa trajanje krivulje. Kakorkoli, s spreminjanjem hitrostnega profila vedno spreminjamo tudi profil višjih odvodov krivulje.

Uporaben problem predstavlja optimizacija krivulje, ki opravi določeno pot v čim krajšem času, pri tem pa ne krši predpisanih omejitev.

Robni pogoji ničte vrste

Ob upoštevanju (6.13) in (6.17) lahko izrazimo tangencialno hitrost v začetni točki kot funkcijo prvih dveh kontrolnih točk:

$$v_0 = \lim_{t \rightarrow t_a^+} v(t) = \frac{b}{\Lambda} \sqrt{(x_{p,1} - x_{p,0})^2 + (y_{p,1} - y_{p,0})^2}, \quad (6.24)$$

Da se pri izračunu smeri krivulje φ , kotne hitrosti ω in tangencialnega pospeška a v začetni robni točki izognemo singularnosti, predpostavimo, da je $v_0 \neq 0$ — govorimo o t. i. robnih pogojih ničte vrste. Ob tem velja omeniti, da izraz (6.24) velja neglede na vrsto začetne robne točke.

Tangencialna hitrost v začetni robni točki (6.24) je različna od nič, ko in samo ko velja $\mathbf{p}_0 \neq \mathbf{p}_1$. V tem primeru lahko s kontrolnimi točkami podamo smer krivulje φ v začetni robni točki:

$$\varphi_0 = \lim_{t \rightarrow t_a^+} \varphi(t) = \arctan \frac{y_{p,1} - y_{p,0}}{x_{p,1} - x_{p,0}} + C\pi, \quad C \in \mathbb{Z}. \quad (6.25)$$

Na sliki 6.2 je prikazan vpliv kontrolnih točk na lastnosti krivulje v začetni robni točki. Koncentrični krogi okoli izhodiščne točke krivulje oz. začetne kontrolne točke $\mathbf{p}_0$ predstavljajo linije konstante tangencialne hitrosti v v začetni robni točki, žarki iz izhodiščne točke pa linije konstantnega izstopnega kota.

Kotno hitrost ω v začetni robni točki podaja enačba:

$$\omega_0 = \lim_{t \rightarrow t_a^+} \omega(t) = \frac{(b-1)}{\Lambda} \frac{(x_{p,1} - x_{p,0})y_{p,2} - (y_{p,1} - y_{p,0})x_{p,2} + x_{p,0}y_{p,1} - x_{p,1}y_{p,0}}{(x_{p,1} - x_{p,0})^2 + (y_{p,1} - y_{p,0})^2}, \quad (6.26)$$

Ugotovimo lahko, da je (6.26) enačba premice, ki je vzporedna s premico, ki poteka skozi kontrolni točki $\mathbf{p}_0$ in $\mathbf{p}_1$ — torej je vzporedna s smerjo krivulje v začetni robni točki. Te premice predstavljajo linije konstantne kotne hitrosti ω v začetni robni točki, kar je prikazano tudi na sliki 6.2. V primeru, ko so točke $\mathbf{p}_0$, $\mathbf{p}_1$ in $\mathbf{p}_2$ kolinearne, postane kotna hitrost v začetni robni točki enaka nič, $\omega_0 = 0$. Iz enačb (6.24) in (6.26) lahko izrazimo tudi ukrivljenost krivulje v začetni robni točki κ_0 , ki je po pričakovanjih neodvisna od časa trajanja trajektorije Λ :

$$\kappa_0 = \lim_{t \rightarrow t_a^+} \kappa(t) = \frac{(b-1)}{b} \frac{(x_{p,1} - x_{p,0})y_{p,2} - (y_{p,1} - y_{p,0})x_{p,2} + x_{p,0}y_{p,1} - x_{p,1}y_{p,0}}{((x_{p,1} - x_{p,0})^2 + (y_{p,1} - y_{p,0})^2)^{\frac{3}{2}}}, \quad (6.27)$$

Pokažemo lahko tudi, da premice, ki so pravokotne na smer krivulje v začetni robni točki, predstavljajo linije konstantnega tangencialnega pospeška a v začetni robni točki. To je razvidno iz naslednje enačbe:

$$a_0 = \lim_{t \rightarrow t_a^+} a(t) = \frac{(b-1)}{\Lambda^2} \frac{(x_{p,2} - 2x_{p,1} + x_{p,0})(x_{p,1} - x_{p,0}) + (y_{p,2} - 2y_{p,1} + y_{p,0})(y_{p,1} - y_{p,0})}{(x_{p,1} - x_{p,0})^2 + (y_{p,1} - y_{p,0})^2}. \quad (6.28)$$

Premica, ki poteka skozi točko $\mathbf{p}_2 = 2\mathbf{p}_0 - \mathbf{p}_1$, predstavlja linijo, kjer je tangencialni pospešek v začetni robni točki enak nič. Ko je izpolnjen pogoj $\mathbf{p}_2 = 2\mathbf{p}_0 - \mathbf{p}_1$ sta kotna hitrost v začetni robni točki ω_0 in tangencialni pospešek v začetni robni točki a_0 enaka nič, kar je posebej predstavljeno na sliki 6.3. Naj ponovimo, s to posebno postavitevjo kontrolnih točk smo dosegli ničelno začetno kotno hitrost ($\omega_0 = 0$) in ničelni začetni tangencialni pospešek ($a_0 = 0$), ob sicer neničelni začetni hitrosti ($v_0 \neq 0$). Pri takšni obravnavi ne smemo pozabiti, da te premice predstavljajo linije konstantne kotne hitrosti in tangencialnega pospeška v začetni robni točki

pri točno določeni medsebojni legi med kontrolnima točkama $\mathbf{p}_0$ in $\mathbf{p}_1$. Torej se s spreminjanjem položaja kontrolnih točk $\mathbf{p}_0$ in $\mathbf{p}_1$ ustrezno spreminjajo tudi linije konstantne kotne hitrosti in tangencialnega pospeška v začetni robni točki. Kontrolne točke zato običajno postavljamo zaporedoma: najprej s postavitvijo začetne kontrolne točke $\mathbf{p}_0$ določimo izhodišče krivulje, nato postavimo kontrolno točko $\mathbf{p}_1$ s katero določimo začetno smer krivulje in začetno tangencialno hitrost, nato s postavitvijo kontrolne točke $\mathbf{p}_2$ vplivamo na želeno začetno kotno hitrost in želeni začetni tangencialni pospešek. S postavitvijo nadaljnjih kontrolnih točk vplivamo na višje odvode v začetni točki, vendar pri tem ne spreminjamo že nastavljenih lastnosti v začetni robni točki.

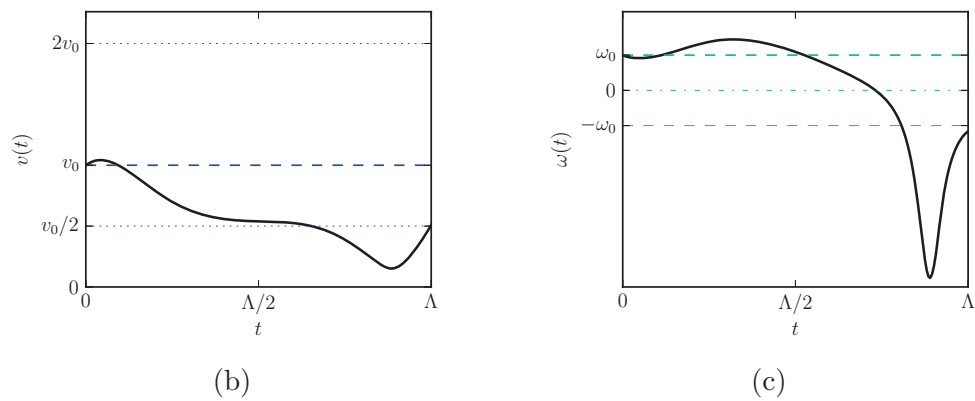
V praksi nas običajno ne zanima, kako kontrolne točke vplivajo na lastnosti krivulje v robnih točkah, temveč želimo postaviti kontrolne točke tako, da zagotovimo določene lastnosti krivulje v robnih točkah. Želene lastnosti krivulje v začetni (izhodiščni) robni točki dosežemo s sledečo postavitvijo kontrolnih točk:

$$\mathbf{p}_0 = \begin{bmatrix} x_0 & y_0 \end{bmatrix}^T, \quad (6.29)$$

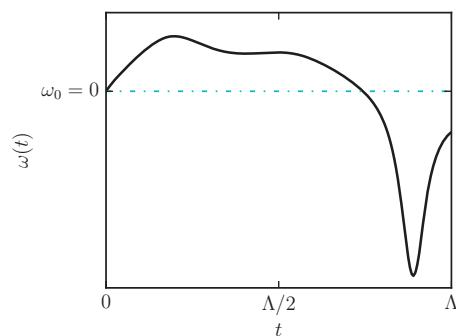
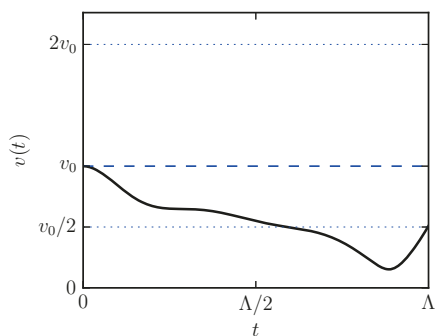
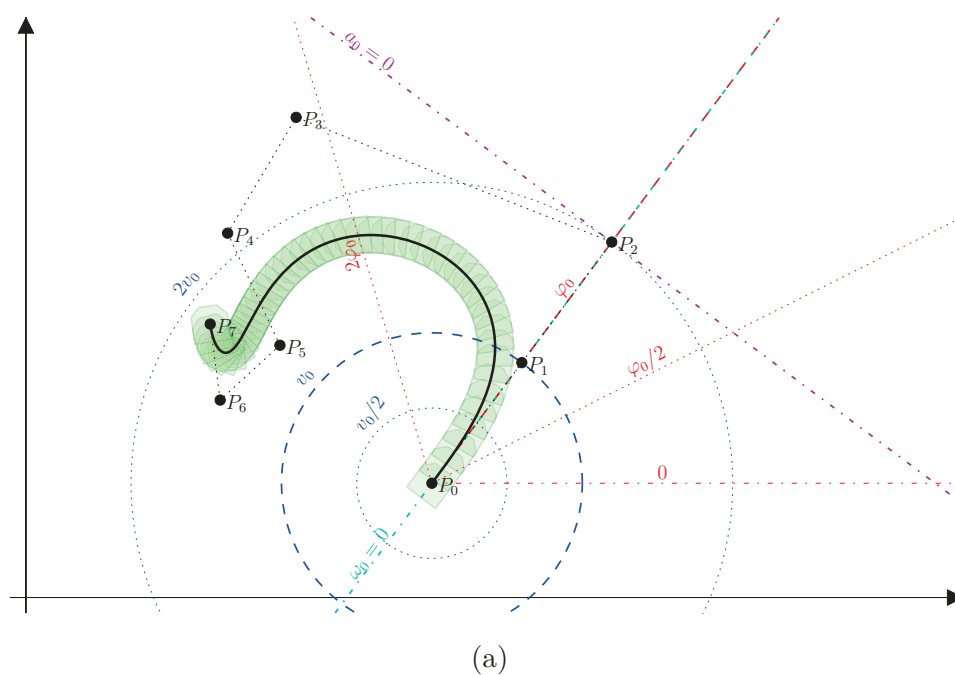
$$\mathbf{p}_1 = \frac{\Lambda v_0}{b} \begin{bmatrix} \cos \varphi_0 & \sin \varphi_0 \end{bmatrix}^T + \mathbf{p}_0, \quad (6.30)$$

$$\mathbf{p}_2 = \frac{\Lambda^2}{b(b-1)} \begin{bmatrix} \cos \varphi_0 & -v_0 \sin \varphi_0 \\ \sin \varphi_0 & v_0 \cos \varphi_0 \end{bmatrix} \begin{bmatrix} a_0 \\ \omega_0 \end{bmatrix} + 2\mathbf{p}_1 - \mathbf{p}_0. \quad (6.31)$$

Kot že omenjeno, se pri tem načrtovanja lotimo tako, da najprej določimo želeno izhodišče (s tem enoznačno določimo prvo kontrolno točko $\mathbf{p}_0$), nato želen izstopni kot in začetno hitrost (s tem enoznačno določimo še drugo kontrolno točko $\mathbf{p}_1$) in nazadnje še želeno kotno hitrost in tangencialni pospešek (tako določimo še tretjo kontrolno točko $\mathbf{p}_2$).



Slika 6.2: (a) Vpliv kontrolnih točk $\mathbf{p}_0$, $\mathbf{p}_1$ in $\mathbf{p}_2$ na lastnosti BB-krivulje v začetni robni točki pri predpostavki, da je začetna tangencialna hitrost različna od nič ($v_0 \neq 0$). Časovni profil (b) tangencialne hitrosti in (c) kotne hitrosti.



Slika 6.3: (a) Posebna postavitev kontrolne točke $\mathbf{p}_2 = 2\mathbf{p}_1 - \mathbf{p}_0$, ki pri predpostavki o neničelni začetni hitrosti $v_0 \neq 0$ oz. $\mathbf{p}_0 \neq \mathbf{p}_1$, zagotovi, da sta tako tangencialni pospešek (vodoraven začetek hitrostnega profila (b)) kot (c) kotna hitrost v začetni robni točki enaka nič.

Robni pogoji prve vrste

Oglejmo si sedaj posebni primer, ko je tangencialna hitrost v začetni točki enaka nič, $v_0 = 0$. Do tega primera pride, ko točki $\mathbf{p}_0$ in $\mathbf{p}_1$ sovpadeta. Zaenkrat predpostavimo, da točka $\mathbf{p}_0 = \mathbf{p}_1$ ne sovpada tudi s točko $\mathbf{p}_2$. V tem primeru enačba (6.19) še vedno drži, vendar pa enačbe (6.20), (6.21), (6.22) in (6.23) postanejo singularne. S pomočjo L'Hôpitalovega pravila lahko z nekaj matematične telovadbe pridemo do smernega vektorja v začetni točki φ_0 :

$$\begin{aligned}\varphi_0 &= \lim_{t \rightarrow t_a^+} \varphi(t) = \frac{\ddot{y}(t)}{\ddot{x}(t)} = \\ &= \arctan \frac{y_{p,2} - y_{p,0}}{x_{p,2} - x_{p,0}} + C\pi, \quad C \in \mathbb{Z},\end{aligned}\quad (6.32)$$

kot tudi kotne hitrosti v začetni točki ω_0 :

$$\begin{aligned}\omega_0 &= \lim_{t \rightarrow t_a^+} \omega(t) = \frac{1}{2} \frac{\ddot{x}(t)\ddot{y}(t) - \dot{\ddot{x}}(t)\dot{\ddot{y}}(t)}{\ddot{x}^2(t) + \ddot{y}^2(t)} = \\ &= \frac{(b-2)}{2\Lambda} \frac{(x_{p,2} - x_{p,0})y_{p,3} - (y_{p,2} - y_{p,0})x_{p,3} + x_{p,0}y_{p,2} - x_{p,2}y_{p,0}}{(x_{p,2} - x_{p,0})^2 + (y_{p,2} - y_{p,0})^2}.\end{aligned}\quad (6.33)$$

Iz enačbe (6.32) je razvidno, da je, v primeru definicije začetne smeri krivulje φ_0 , kontrolna točka $\mathbf{p}_2$ nadomestila vlogo kontrolne točke $\mathbf{p}_1$. Razdalja med točkama $\mathbf{p}_2$ in $\mathbf{p}_0$ sedaj določa velikost tangencialnega pospeška v začetni točki:

$$\begin{aligned}a_0 &= \lim_{t \rightarrow t_a^+} a(t) = \sqrt{\ddot{x}^2(t) + \ddot{y}^2(t)} = \\ &= \frac{b(b-1)}{\Lambda^2} \sqrt{(x_{p,2} - x_{p,0})^2 + (y_{p,2} - y_{p,0})^2}.\end{aligned}\quad (6.34)$$

Enačba (6.33) ima enako obliko kot enačba (6.26), le da sta točki $\mathbf{p}_1$ in $\mathbf{p}_2$ zamenjani s točkami $\mathbf{p}_2$ in $\mathbf{p}_3$. Premice, ki so vzporedne s premico, ki poteka skozi kontrolni točki $\mathbf{p}_0$ in $\mathbf{p}_2$, zatorej zopet predstavljajo linije konstantne kotne hitrosti v začetni točki. Premice, ki so pravokotne na premico, ki poteka skozi kontrolni točki $\mathbf{p}_0$ in $\mathbf{p}_2$, so verjetno linije konstantnega odvoda tangencialnega pospeška $\dot{a}_0$. Predpostavka, da kontrolna točka $\mathbf{p}_2$ ne sovpada s kontrolno točko $\mathbf{p}_0 = \mathbf{p}_1$, pomeni, da je tangencialni pospešek v začetni točki različen od nič, $a_0 \neq 0$, kar je razvidno iz enačbe (6.34).

Želene lastnosti krivulje v začetni robni točki dosežemo z naslednjo postavitvijo

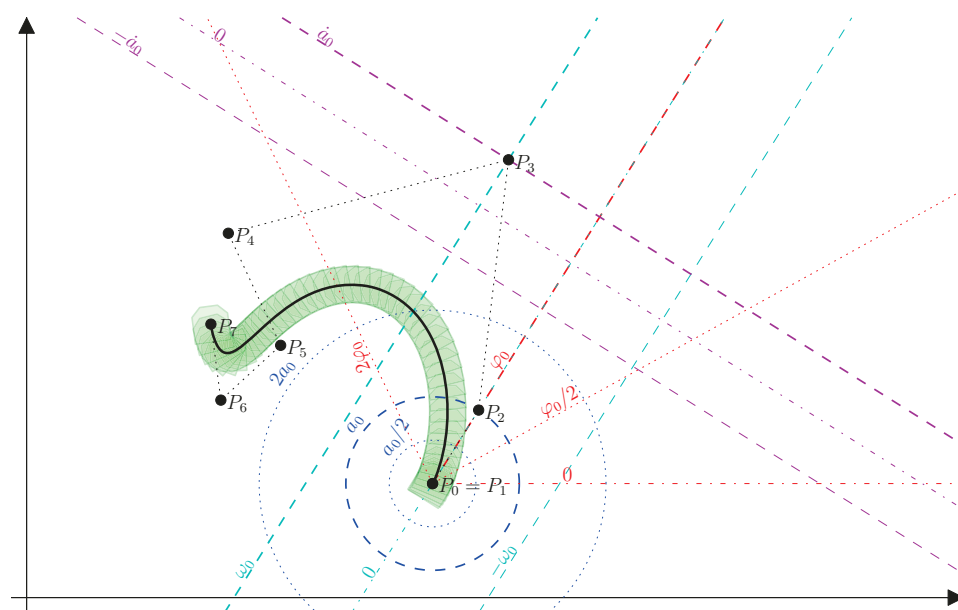
kontrolnih točk:

$$\mathbf{p}_0 = \mathbf{p}_1 = \begin{bmatrix} x_0 & y_0 \end{bmatrix}^T, \quad (6.35)$$

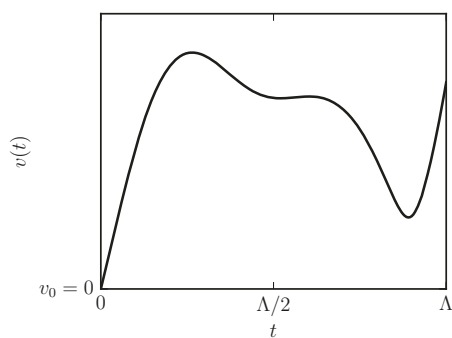
$$\mathbf{p}_2 = \frac{\Lambda^2 a_0}{b(b-1)} \begin{bmatrix} \cos \varphi_0 & \sin \varphi_0 \end{bmatrix}^T + \mathbf{p}_0, \quad (6.36)$$

$$\mathbf{p}_3 = \frac{\Lambda^3}{b(b-1)(b-2)} \begin{bmatrix} \cos \varphi_0 & -2a_0 \sin \varphi_0 \\ \sin \varphi_0 & 2a_0 \cos \varphi_0 \end{bmatrix} \begin{bmatrix} \dot{a}_0 \\ \omega_0 \end{bmatrix} + 3\mathbf{p}_2 - 2\mathbf{p}_0. \quad (6.37)$$

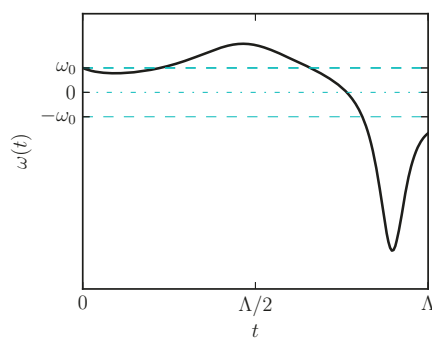
Omeniti velja še, da izračunane limite — prve vrstice v enačbah (6.32), (6.33) in (6.34) — ne veljajo le za BB-krivulje, temveč veljajo splošno za vse krivulje, za katere obstajajo limite odvodov, ki nastopajo v enačbah. Prav tako so splošni tudi izrazi (6.19), (6.20), (6.21), (6.22) in (6.23).



(a)

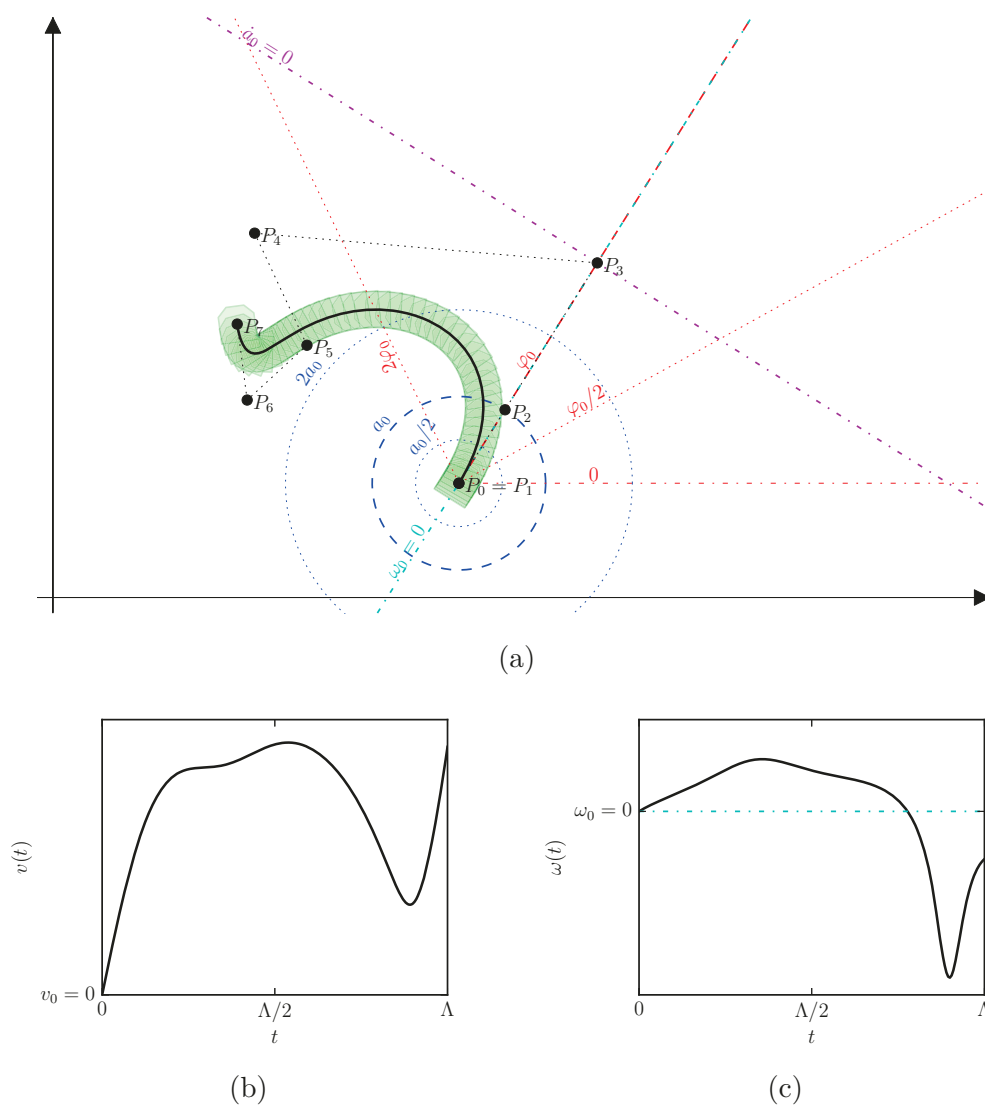


(b)



(c)

Slika 6.4: (a) Vpliv kontrolnih točk $\mathbf{p}_0$, $\mathbf{p}_2$ in $\mathbf{p}_3$ na lastnosti BB-krivulje v začetni robni točki pri predpostavki $v_0 = 0 \wedge a_0 \neq 0$. Časovni profil (b) tangencialne hitrosti in (c) kotne hitrosti.



Slika 6.5: (a) Posebna postavitev kontrolne točke $\mathbf{p}_3 = 3\mathbf{p}_2 - 2\mathbf{p}_0$, ki pri predpostavki $v_0 = 0 \wedge a_0 \neq 0$ oz. $\mathbf{p}_0 = \mathbf{p}_1 \neq \mathbf{p}_2$, zagotovi, da sta tako prvi odvod tangencialnega pospeška (vpliva na obliko tangencialne hitrosti (b)) kot (c) kotna hitrost v začetni robni točki enaka nič.

Robni pogoji druge vrste

V prejšnjem poglavju je prikazano, kako je možno načrtati BB-krivuljo z ničelno tangencialno hitrostjo v robni točki. Pri tem smo zahtevali neničelni tangencialni pospešek v robni točki. Sedaj pa pokažimo, da je moč načrtati tudi krivuljo z ničelno tangencialno hitrostjo in ničelnim tangencialnim pospeškom v robni točki. Zopet obravnavajmo le stanje v začetni robni točki in zahtevajmo neničelni prvi odvod tangencialnega pospeška v začetni robni točki. Obravnavamo torej primer $v_0 = 0 \wedge a_0 = 0 \wedge \dot{a}_0 \neq 0$, ki ga lahko v primeru BB-krivulj enakovredno predstavimo s pogojem $\mathbf{p}_0 = \mathbf{p}_1 = \mathbf{p}_2 \neq \mathbf{p}_3$. Slika postavitve začetnih točk je v tem primeru prikazana na sliki 6.6.

Izstopni kot φ_0 je v tem primeru podan z enačbo:

$$\begin{aligned}\varphi_0 &= \lim_{t \rightarrow t_a^+} \varphi(t) = \arctan \frac{\ddot{y}(t)}{\ddot{x}(t)} = \\ &= \arctan \frac{y_{p,3} - y_{p,0}}{x_{p,3} - x_{p,0}} + C\pi, \quad C \in \mathbb{Z},\end{aligned}\tag{6.38}$$

torej sedaj, pri fiksni kontrolni točki $\mathbf{p}_0$, kontrolna točka $\mathbf{p}_3$ določa smer izstopnega kota φ_0 v začetni robni točki. Kontrolna točka $\mathbf{p}_3$ je torej nadomestila vlogo določanja smeri krivulje v začetni robni točki, ki jo je imela kontrolna točka $\mathbf{p}_2$ v primeru robnih pogojev prve vrste oz. kontrolna točka $\mathbf{p}_1$ v primeru robnih pogojev ničte vrste. Koncentrični krogi okoli izhodiščne kontrolne točke $\mathbf{p}_0$ sedaj predstavljajo linije konstantnega prvega odvoda tangencialnega pospeška $\dot{a}$ v začetni robni točki, kar je razvidno iz naslednje enačbe:

$$\begin{aligned}\dot{a}_0 &= \lim_{t \rightarrow t_a^+} \dot{a}(t) = \sqrt{\ddot{x}^2(t) + \ddot{y}^2(t)} = \\ &= \frac{b(b-1)(b-2)}{\Lambda^3} \sqrt{(x_{p,3} - x_{p,0})^2 + (y_{p,3} - y_{p,0})^2}.\end{aligned}\tag{6.39}$$

Kontrolna točka $\mathbf{p}_3$ vpliva na prvi odvod tangencialne hitrosti v začetni robni točki na podoben način, kot je kontrolna točka $\mathbf{p}_2$ ($\mathbf{p}_1$) vplivala na tangencialni pospešek (tangencialno hitrost) v začetni robni točki v primeru robnih pogojev prve (ničte) vrste.

Tudi v tem primeru poiščemo še limito kotne hitrosti ω v začetni robni točki:

$$\omega_0 = \lim_{t \rightarrow t_a^+} \omega(t) = \frac{1}{3} \frac{\ddot{x}(t)y^{(iv)}(t) - x^{(iv)}(t)\ddot{y}(t)}{\ddot{x}^2(t) + \ddot{y}^2(t)} = \quad (6.40)$$

$$= \frac{b-3}{\Lambda} \frac{(x_{p,3} - x_{p,0})y_{p,4} - (y_{p,3} - y_{p,0})x_{p,4} + x_{p,0}y_{p,3} - x_{p,3}y_{p,0}}{(x_{p,3} - x_{p,0})^2 + (y_{p,3} - y_{p,0})^2}, \quad (6.41)$$

kar nas zoper pripelje do enačbe premice, ki v tem primeru poteka skozi kontrolni točki $\mathbf{p}_0$ in $\mathbf{p}_3$. Premice vzporedne na linije konstantne kotne hitrosti v začetni točki ω_0 sedaj predstavljajo linije konstantnega drugega odvoda tangencialne hitrosti v začetni robni točki, kar je razvidno iz:

$$\begin{aligned} \ddot{a}_0 = \lim_{t \rightarrow t_a^+} \ddot{v}(t) &= \frac{\ddot{x}(t)x^{(iv)}(t) + \ddot{y}(t)y^{(iv)}(t)}{\sqrt{\ddot{x}^2(t) + \ddot{y}^2(t)}} = \frac{b(b-1)(b-2)(b-3)}{\Lambda^4} \\ &\frac{(x_{p,3} - x_{p,0})(x_{p,4} - 4x_{p,3} + 3x_{p,0}) + (y_{p,3} - y_{p,0})(y_{p,4} - 4y_{p,3} + 3y_{p,0})}{\sqrt{(x_{p,3} - x_{p,0})^2 + (y_{p,3} - y_{p,0})^2}}. \end{aligned} \quad (6.42)$$

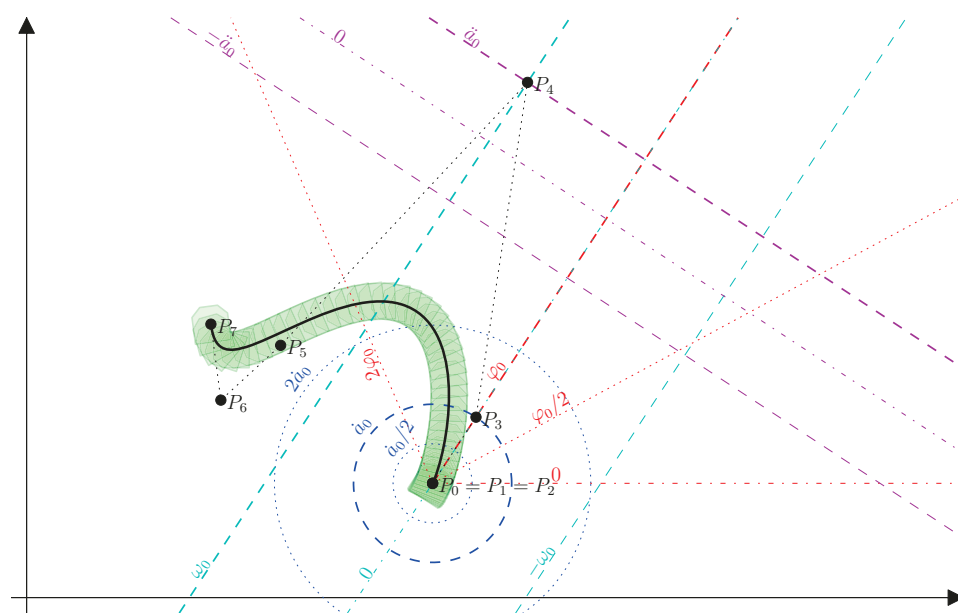
Pri fiksni postavitvi kontrolnih točk $\mathbf{p}_0$ in $\mathbf{p}_3$, vplivamo s kontrolno točko $\mathbf{p}_4$ na kotno hitrost in drugi odvod tangencialnega pospeška v začetni robni točki. S posebno postavitvijo kontrolne točke $\mathbf{p}_4 = 4\mathbf{p}_3 - 3\mathbf{p}_0$ dosežemo, da sta tako kotna hitrost v začetni robni točki kot drugi odvod tangencialnega pospeška v začetni robni točki enaka nič — ta postavitev je posebej prikazana na sliki 6.7.

Predstavimo še, kako postavimo kontrolne točke, da dosežemo želene lastnosti krivulje v začetni robni točki:

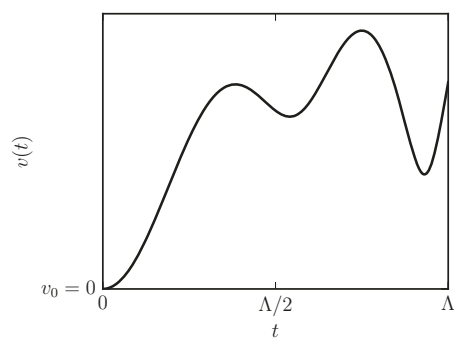
$$\mathbf{p}_0 = \mathbf{p}_1 = \mathbf{p}_2 = \begin{bmatrix} x_0 & y_0 \end{bmatrix}^T, \quad (6.43)$$

$$\mathbf{p}_3 = \frac{\Lambda^3 \ddot{a}_0}{b(b-1)(b-2)} \begin{bmatrix} \cos \varphi_0 & \sin \varphi_0 \end{bmatrix}^T + \mathbf{p}_0, \quad (6.44)$$

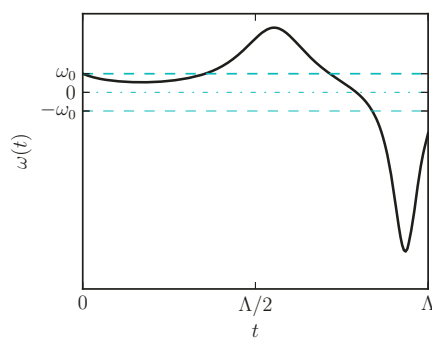
$$\mathbf{p}_4 = \frac{\Lambda^4}{b(b-1)(b-2)(b-3)} \begin{bmatrix} \cos \varphi_0 & -3\ddot{a}_0 \sin \varphi_0 \\ \sin \varphi_0 & 3\ddot{a}_0 \cos \varphi_0 \end{bmatrix} \begin{bmatrix} \ddot{a}_0 \\ \omega_0 \end{bmatrix} + 4\mathbf{p}_3 - 3\mathbf{p}_0. \quad (6.45)$$



(a)

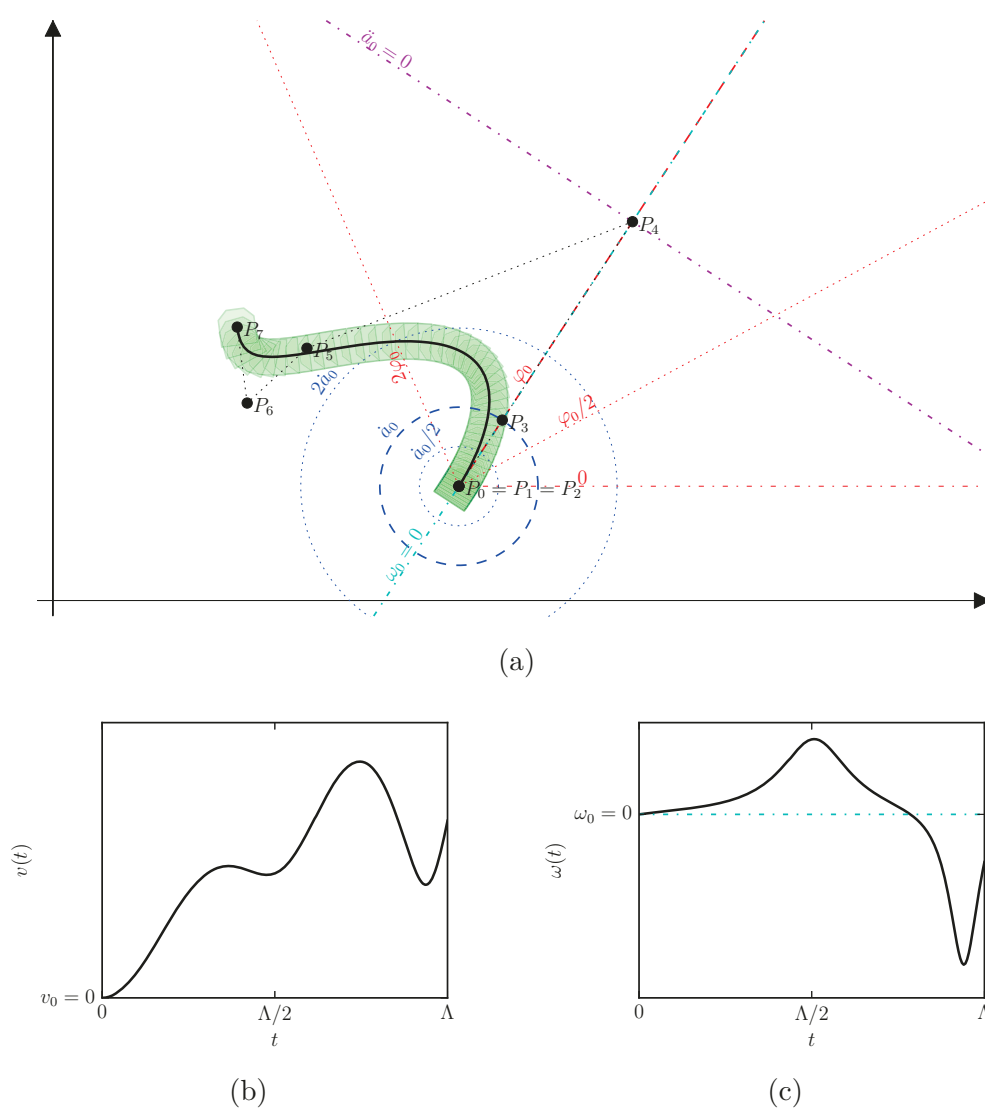


(b)



(c)

Slika 6.6: (a) Vpliv kontrolnih točk $\mathbf{p}_0$, $\mathbf{p}_3$ in $\mathbf{p}_4$ na lastnosti BB-krivulje v začetni robni točki pri predpostavki $v_0 = 0 \wedge a_0 = 0 \wedge \dot{a}_0 \neq 0$. Časovni profil (b) tangencialne hitrosti in (c) kotne hitrosti.



Slika 6.7: (a) Posebna postavitev kontrolne točke $\mathbf{p}_4 = 4\mathbf{p}_3 - 3\mathbf{p}_0$, ki pri predpostavki $v_0 = 0 \wedge a_0 = 0 \wedge \dot{a}_0 \neq 0$ oz. $\mathbf{p}_0 = \mathbf{p}_1 = \mathbf{p}_2 \neq \mathbf{p}_3$, zagotovi, da sta tako drugi odvod tangencialnega pospeška (vpliva na obliko tangencialne hitrosti (b)) kot (c) kotna hitrost v začetni robni točki enaka nič.

Robni pogoji višjih vrst

Obravnavo robnih pogojev bi lahko nadaljevali tudi na višje vrste, vendar pa takšna obravnava nima velike praktične uporabnosti. Vsaka nadaljnja lastnost višjega reda v končni točki poveča število začetnih/končnih kontrolnih točk, ki le-to lastnost določajo, s čimer se efektivno viša red celotne krivulje, če želimo lastnosti v začetni robni točki določiti neodvisno od lastnosti v končni robni točki.

6.2.2 Zlepki

S primerno visokim redom BB-krivulje lahko opišemo krivuljo poljubne oblike s poljubno natančnim približkom, vendar se pri višjih redih pojavi problem numerične stabilnosti. Zato je bolje opisati poljubno pot z zlepkom več BB-krivulj nižjega reda. Zlepek BB-krivulj nam omogoča, da potegnemo krivuljo skozi množico točk. Zlepek lahko zgradimo na več načinov, odvisno od predpisanih pogojev, ki jim želimo zadostiti v robnih oz. povezovalnih točkah — kar določa končni razred odvedljivosti zlepka. V nadaljevanju so predstavljeni trije načini lepljenja krivulj, kjer zahtevamo le zvezen zlepek (zlepek razreda $\mathcal{C}^0$), zvezen zlepek z zveznim prvim odvodom (zlepek razreda $\mathcal{C}^1$) in zvezen zlepek z zveznimi odvodi do vključno drugega reda (zlepek razreda $\mathcal{C}^2$). Predstavljeno je, kako se zahteve o zveznosti odražajo v lastnostih v robnih točkah, kot so: končna/začetna točka, izstopni/vstopni kot, končna/začetna tangencialna hitrost, končna/začetna kotna hitrost itd. Obravnavamo zlepek dveh zaporednih BB-trajektorij $\mathbf{r}_{\lambda,j}(\lambda)$ in $\mathbf{r}_{\lambda,j+1}(\lambda)$. Predpostavimo, da sta časa trajanja Λ_j in Λ_{j+1} v splošnem različna. Zaradi preprostejšje notacije, predpostavimo, da je čas t na začetku vsake krivulje v zlepku enak nič. Predpostavimo še, da so vse krivulje v zlepku enakega reda b , čeprav bi lahko bile tudi različnega reda.

Zlepek razreda $\mathcal{C}^0$

Da dosežemo zveznost zlepka, moramo zadostiti naslednjemu pogoju:

$$\lim_{\lambda \rightarrow 1^-} \mathbf{r}_{\lambda,j}(\lambda) = \lim_{\lambda \rightarrow 0^+} \mathbf{r}_{\lambda,j+1}(\lambda), \quad (6.46)$$

ki pomeni, da se konec krivulje j stika z začetkom krivulje $j + 1$. Zahtevi (6.46) zadostimo, ko prva kontrolna točka krivulje $j + 1$ sovпада z zadnjo kontrolno

točko krivulje j , kar podaja izraz:

$$\mathbf{p}_{0,j+1} = \mathbf{p}_{b,j}. \quad (6.47)$$

S tem smo očitno dosegli, da konec krivulje j sovpada z začetkom krivulje $j + 1$, vendar pa vstopni kot v zadnjo točko krivulje j ni enak izstopnemu kotu iz začetne točke na krivulji $j + 1$.

Zlepek razreda $\mathcal{C}^1$

Da dosežemo zvezen zlepek z zveznimi prvimi odvodi, moramo poleg zahteve (6.46) zadostiti tudi sledeči zahtevi:

$$\frac{1}{\Lambda_j} \lim_{\lambda \rightarrow 1^-} \frac{d\mathbf{r}_{\lambda,j}(\lambda)}{d\lambda} = \frac{1}{\Lambda_{j+1}} \lim_{\lambda \rightarrow 0^+} \frac{d\mathbf{r}_{\lambda,j+1}(\lambda)}{d\lambda}, \quad (6.48)$$

kar dosežemo s sledečo postavitvijo druge kontrolne točke na krivulji $j + 1$:

$$\mathbf{p}_{1,j+1} = \left(\frac{\Lambda_{j+1}}{\Lambda_j} + 1 \right) \mathbf{p}_{b,j} - \frac{\Lambda_{j+1}}{\Lambda_j} \mathbf{p}_{b-1,j}, \quad (6.49)$$

seveda pa moramo pri tem zadostiti tudi (6.47). Pogoj (6.48) je zadosten le v primeru, da je tangencialna hitrost v končni robni točki krivulje j različna od nič $v_{b,j} = v_{0,j+1} \neq 0$, sicer moramo zadostiti dodatnim pogojem. V primeru $v_{b,j} = v_{0,j+1} = 0 \wedge a_{b,j} = a_{0,j+1} \neq 0$, pridemo do dodatnega pogoja:

$$\mathbf{p}_{2,j+1} = \left(\frac{\Lambda_{j+1}}{\Lambda_j} + 1 \right) \mathbf{p}_{b,j} - \frac{\Lambda_{j+1}}{\Lambda_j} \mathbf{p}_{b-2,j}. \quad (6.50)$$

Izpeljati je možno tudi pogoje za primer $v_{b,j} = v_{0,j+1} = 0 \wedge a_{b,j} = a_{0,j+1} = 0 \wedge \dot{a}_{b,j} = \dot{a}_{0,j+1} \neq 0$ in tudi ostale posebne primere. Na tem mestu velja omeniti še, da zveznost zlepka v primeru ničelne tangencialne hitrosti v povezovalni točki lahko dosežemo tudi z vpeljavo dodatne vmesne krivulje, ki s primernim vrtenjem v točki zadosti zahtevi (6.48).

Zlepek razreda $\mathcal{C}^2$

Da dosežemo zveznost tudi drugega odvoda, moramo poleg zahtev (6.51) in (6.48) zadostiti še naslednji zahtevi:

$$\frac{1}{\Lambda_j^2} \lim_{\lambda \rightarrow 1^-} \frac{d^2 \mathbf{r}_{\lambda,j}(\lambda)}{d\lambda^2} = \frac{1}{\Lambda_{j+1}^2} \lim_{\lambda \rightarrow 0^+} \frac{d^2 \mathbf{r}_{\lambda,j+1}(\lambda)}{d\lambda^2}, \quad (6.51)$$

ki jo izpolnimo z naslednjo postavitvijo tretje kontrolne točke na krivulji $j + 1$:

$$\mathbf{p}_{2,j+1} = \left(\frac{\Lambda_{j+1}}{\Lambda_j} + 1 \right)^2 \mathbf{p}_{b,j} - 2 \frac{\Lambda_{j+1}}{\Lambda_j} \left(\frac{\Lambda_{j+1}}{\Lambda_j} + 1 \right) \mathbf{p}_{b-1,j} + \frac{\Lambda_{j+1}^2}{\Lambda_j^2} \mathbf{p}_{b-2,j}. \quad (6.52)$$

Na tem mestu moramo zopet opozoriti, da je pogoj (6.52) zadosten le, če je tangencialna hitrost v končni robni točki krivulje j različna od nič $v_{b,j} = v_{0,j+1} \neq 0$. Sicer moramo postopati podobno kot v prejšnjem primeru in zadostiti dodatnim pogojem oz. vpeljati dodatno vmesno krivuljo, ki s primernim vrtenjem na mestu zadosti zahtevi (6.51).

Povzetek

V primeru, da tangencialna hitrost v povezovalni točki ni enaka nič ($v_{b,j} \neq 0$), so pogoji (6.47) do (6.52) zadostni za zveznost položaja, smernega kota, tangencialne hitrosti, kotne hitrosti in tangencialnega pospeška. V singularnem primeru, ko je tangencialna hitrost v povezovalni točki enaka nič, moramo zadostiti dodatnim pogojem, da dosežemo želeno stopnjo zveznosti. Z zlepkami smo dosegli, da lahko predpišemo točke, skozi katere želimo, da krivulja poteka, poleg tega pa lahko v teh predpisanih točkah izrazimo dodatne želje glede usmerjenosti krivulje, tangencialne hitrosti in pospeška, kotne hitrosti itd.

6.2.3 Optimizacija poti

S skaliranjem časovne funkcije, kot je opisano v poglavju 6.1.1, lahko izdelamo trajektorijo, ki zadosti omejitvam (6.4). Kot je bilo omenjeno in prikazano, je takšen način optimizacije vedno mogoč. Vendar pa nas sedaj zanima, ali je možno optimizirati hitrostni in/ali pospeškovni profil trajektorije, brez spreminjanja dolžine trajanja trajektorije in/ali brez spreminjanja določenih lastnosti v robnih točkah krivulje (npr. začetne/končne tangencialne/kotne hitrosti/pospeška). Pri tem naj omenimo, da veličin, ki so v robnih točkah enake nič, s skaliranjem le-teh ne spreminjamo — torej v tem primeru avtomatično zadostimo zahtevi po invariantnosti teh lastnosti v robnih točkah, a vseeno se v tem primeru spremeni dolžina trajanja trajektorije.

V nadaljevanju so obravnavane le na BB-krivulje, ki jih enolično določa množica kontrolnih točk in izbrana dolžina trajanja trajektorije. Končno število začetnih oz.

končnih kontrolnih točk določa lastnosti krivulje v robnih točkah, kot je opisano v poglavju 6.2.1. Preostale vmesne točke lahko uporabimo za spreminjanje oblike krivulje, s čimer ne vplivamo na lastnosti krivulje v robnih točkah, temveč le na obliko krivulje in na lastnosti krivulje med robnima točkama. Našo obravnavo bomo omejili na red BB-krivulj, ki imajo eno dodatno vmesno kontrolno točko, ki jo poimenujemo prosta kontrolna točka. S premikanjem proste kontrolne točke $\mathbf{p}_\star$ po dvorazsežnem prostoru torej lahko vplivamo na obliko krivulje in njene lastnosti v nerobnih točkah, pri tem pa se dolžina trajanja trajektorije ne spremeni — prav tako se ne spremenijo določene lastnosti v robnih točkah, ki so določene z začetnimi/končnimi kontrolnimi točkami.

Zopet naletimo na optimizacijski problem, kjer iščemo optimalni položaj proste kontrolne točke, ki minimizira vrednost določene kriterijske funkcije, ki je odvisna od cilja optimizacije. Cilje optimizacije je lahko minimizacija maksimalne vrednosti tangencialne/kotne hitrosti/pospeška kot tudi minimizacija dolžine poti, minimizacija maksimalne ukrivljenosti krivulje itd. Pogoste kriterijske funkcije so torej:

$$J_s(\mathbf{p}_\star) = \int_0^1 v_\lambda(\lambda, \mathbf{p}_\star) d\lambda, \quad (6.53)$$

$$J_z(\mathbf{p}_\star) = \int_0^1 v_\lambda^2(\lambda, \mathbf{p}_\star) d\lambda, \quad (6.54)$$

$$J_\kappa(\mathbf{p}_\star) = \max_\lambda |\kappa_\lambda(\lambda, \mathbf{p}_\star)|, \quad (6.55)$$

$$J_v(\mathbf{p}_\star) = \max_\lambda |v_\lambda(\lambda, \mathbf{p}_\star)|, \quad (6.56)$$

$$J_\omega(\mathbf{p}_\star) = \max_\lambda |\omega_\lambda(\lambda, \mathbf{p}_\star)|, \quad (6.57)$$

$$J_a(\mathbf{p}_\star) = \max_\lambda |a_\lambda(\lambda, \mathbf{p}_\star)|, \quad (6.58)$$

$$J_\alpha(\mathbf{p}_\star) = \max_\lambda |\alpha_\lambda(\lambda, \mathbf{p}_\star)|, \quad (6.59)$$

kjer je λ omejena na zaprti interval $\lambda \in [0, 1]$. Najprej si oglejmo nekaj primerov, ko optimiziramo le eno izmed kriterijskih funkcij (6.53) do (6.59), kasneje pa tudi primer optimizacije več kriterijskih funkcij hkrati.

Cilj optimizacije je poiskati takšno vrednost proste kontrolne točke $\mathbf{p}_\star^T =$

$[x_{p,\star} \ y_{p,\star}]$, ki doseže minimalno vrednost kriterijske funkcije:

$$\mathbf{p}_{\star, opt.} = \arg \min_{\mathbf{p}_{\star}} J(\mathbf{p}_{\star}) . \quad (6.60)$$

Domena Ω iskanja optimalne vrednosti je dvorazsežni prosto realnih števil $\Omega \in \mathbb{R} \times \mathbb{R}$, saj vsebuje prosta kontrolna točka dva elementa, $\mathbf{p}_{\star} \in \mathbb{R}^2$.

Podrobneje si oglejmo primer, ko želimo poiskati najkrajšo pot med začetno in končno točko. Optimizacijski problem se torej glasi:

$$\mathbf{p}_{\star, opt. \ s} = \arg \min_{\mathbf{p}_{\star}} J_s(\mathbf{p}_{\star}) . \quad (6.61)$$

Optimizacijski problem lahko rešujemo s primerno optimizacijsko metodo. Čeprav lahko optimizacijski problem (6.61) rešujemo z optimizacijskimi metodami, ki ne zahtevajo poznavanje gradienta kriterijske funkcije, lahko ob znanem gradientu optimizacijski problem rešimo bolj učinkovito z manj izračuni kriterijske funkcije. Poiščimo torej gradient funkcije (6.53). Očitno je, da ima odvod krivulje $\mathbf{r}_{\lambda}(\lambda) = [x_{\lambda}(\lambda) \ y_{\lambda}(\lambda)]$ obliko:

$$\begin{aligned} \frac{dx_{\lambda}(\lambda)}{d\lambda} &= A_x(\lambda)x_{p,\star} + B_x(\lambda) , \\ \frac{dy_{\lambda}(\lambda)}{d\lambda} &= A_y(\lambda)y_{p,\star} + B_y(\lambda) . \end{aligned} \quad (6.62)$$

Torej lahko kriterijsko funkcijo (6.53), ki nastopa v optimizacijskem problemu (6.61) zapišemo, kot sledi:

$$J_s(\mathbf{p}_{\star}) = \int_0^1 \sqrt{(A_x(\lambda)x_{p,\star} + B_x(\lambda))^2 + (A_y(\lambda)y_{p,\star} + B_y(\lambda))^2} d\lambda . \quad (6.63)$$

Označimo izraz pod korenem v enačbi (6.63) z $\beta(\lambda, \mathbf{p}_{\star}) = (A_x(\lambda)x_{p,\star} + B_x(\lambda))^2 +$

$(A_y(\lambda)y_{p,\star} + B_y(\lambda))^2$. Gradient kriterijske funkcije (6.53) je torej:

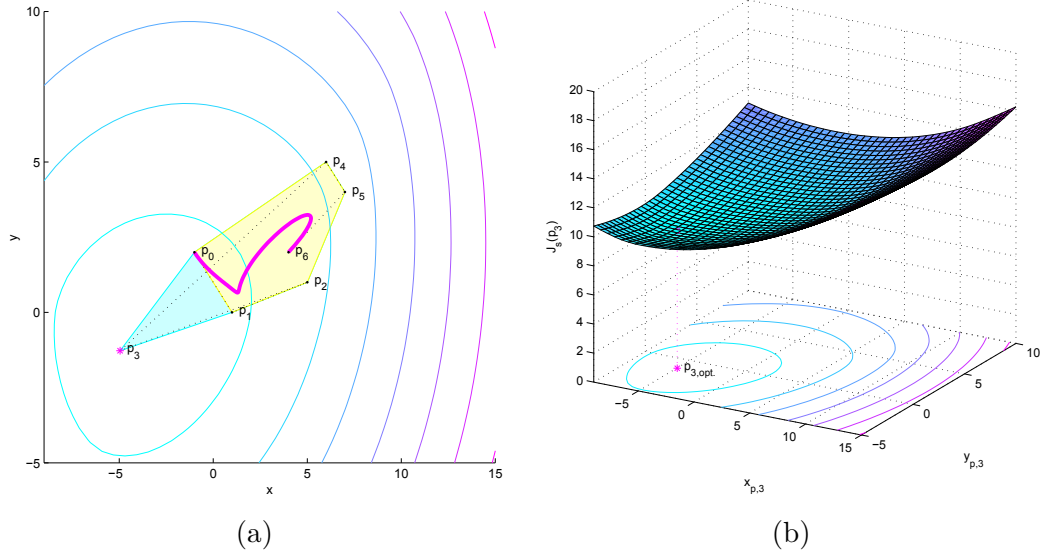
$$\frac{\partial J_s(\mathbf{p}_\star)}{\partial \mathbf{p}_\star} = \frac{\partial}{\partial \mathbf{p}_\star} \int_0^1 \sqrt{\beta(\lambda, \mathbf{p}_\star)} d\lambda = \int_0^1 \frac{\partial}{\partial \mathbf{p}_\star} \sqrt{\beta(\lambda, \mathbf{p}_\star)} d\lambda = \quad (6.64)$$

$$= \int_0^1 \frac{1}{\sqrt{\beta(\lambda, \mathbf{p}_\star)}} \begin{bmatrix} A_x^2(\lambda)x_{p,\star} + A_x(\lambda)B_x(\lambda) \\ A_y^2(\lambda)y_{p,\star} + A_y(\lambda)B_y(\lambda) \end{bmatrix} d\lambda = \quad (6.65)$$

$$= \mathbf{p}_\star^T \int_0^1 \frac{1}{\sqrt{\beta(\lambda, \mathbf{p}_\star)}} \begin{bmatrix} A_x^2(\lambda) & 0 \\ 0 & A_y^2(\lambda) \end{bmatrix} d\lambda + \quad (6.66)$$

$$+ \int_0^1 \frac{1}{\sqrt{\beta(\lambda, \mathbf{p}_\star)}} \begin{bmatrix} A_x(\lambda)B_x(\lambda) \\ A_y(\lambda)B_y(\lambda) \end{bmatrix} d\lambda. \quad (6.67)$$

Optimizacijski problem lahko torej rešimo s kvazi-Newtonovo optimizacijsko metodo. Na sliki 6.8a je prikazan primer rezultata optimizacije, kjer smo poiskali najkrajšo pot med začetno in končno točko. Na sliki 6.8b je prikazana še kriterijska funkcija (6.53).



Slika 6.8: Najkrajša pot pri podanih zahtevah v robnih točkah in podanim časom trajanja trajektorije (optimizacija s prosto vmesno točko).

Oglejmo si sedaj še kriterijsko funkcijo (6.54), ki opisuje integral kvadratov tangencialne hitrosti po krivulji. V tem primeru se optimizacijski problem glasi:

$$\mathbf{p}_{\star, opt.} = \arg \min_{\mathbf{p}_\star} J_z(\mathbf{p}_\star). \quad (6.68)$$

Izkaže se, da je rešitev optimizacijskega problema (6.61) možno zapisati v eksplisicni obliki, ki pa žal ni analitično rešljiva. Kakorkoli, rešitev je možno enostavno poiskati z numerično integracijo. Na tem mestu naj omenimo še, da rešitve optimizacijskega problema iskanja najkrajše poti ni možno zapisati v eksplisicni obliki, kar je razvidno iz enačbe (6.67), saj ni možno izpostaviti člena $\beta(\lambda, \mathbf{p}_\star)$ izpod integrala. Z upoštevanjem relacije (6.62) zapišemo kriterijsko funkcijo (6.54), ki nastopa v optimizacijskem problemu (6.68), kot sledi:

$$J_z(\mathbf{p}_\star) = \int_0^1 (A_x(\lambda)x_{p,\star} + B_x(\lambda))^2 + (A_y(\lambda)y_{p,\star} + B_y(\lambda))^2 d\lambda. \quad (6.69)$$

Kritične točke funkcije (6.69) poiščemo s pomočjo gradienta

$$\frac{\partial J_z(\mathbf{p}_\star)}{\partial \mathbf{p}_\star} = \frac{\partial}{\partial \mathbf{p}_\star} \int_0^1 \beta(\lambda, \mathbf{p}_\star) d\lambda = \int_0^1 \frac{\partial}{\partial \mathbf{p}_\star} \beta(\lambda, \mathbf{p}_\star) d\lambda = \quad (6.70)$$

$$= \int_0^1 \begin{bmatrix} A_x^2(\lambda)x_{p,\star} + A_x(\lambda)B_x(\lambda) \\ A_y^2(\lambda)y_{p,\star} + A_y(\lambda)B_y(\lambda) \end{bmatrix} d\lambda = \quad (6.71)$$

$$= \mathbf{p}_\star^T \int_0^1 \begin{bmatrix} A_x^2(\lambda) & 0 \\ 0 & A_y^2(\lambda) \end{bmatrix} d\lambda + \int_0^1 \begin{bmatrix} A_x(\lambda)B_x(\lambda) \\ A_y(\lambda)B_y(\lambda) \end{bmatrix} d\lambda. \quad (6.72)$$

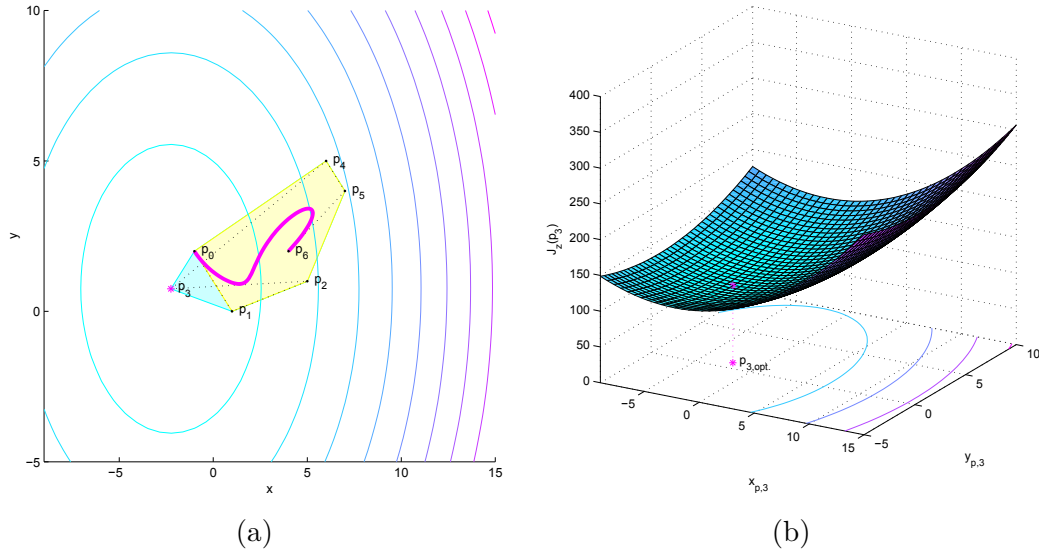
V kritičnih točkah je gradient enak nič ($\frac{\partial J_z(\mathbf{p}_\star)}{\partial \mathbf{p}_\star} = \mathbf{0}$), kar nam omogoča določitev kritične točke:

$$\mathbf{p}_{\star, \text{opt. } z}^T = - \begin{bmatrix} \frac{\int_0^1 A_x(\lambda)B_x(\lambda) d\lambda}{\int_0^1 A_x^2(\lambda) d\lambda} & \frac{\int_0^1 A_y(\lambda)B_y(\lambda) d\lambda}{\int_0^1 A_y^2(\lambda) d\lambda} \end{bmatrix}. \quad (6.73)$$

Zadosten pogoj za lokalni minimum je, da je determinanta Hessejeve matrike večja od nič in da je drugi parcialni odvod kriterijske funkcije v kritični točki večji od nič. Integrali v enačbi (6.73) niso analitično rešljivi, a jih lahko rešimo z numeričnim približkom:

$$\mathbf{p}_{\star, \text{opt. } z}^T \approx - \begin{bmatrix} \frac{\sum_{i=0}^N A_x(\lambda_i)B_x(\lambda_i)}{\sum_{i=1}^N A_x^2(\lambda_i)} & \frac{\sum_{i=0}^N A_y(\lambda_i)B_y(\lambda_i)}{\sum_{i=1}^N A_y^2(\lambda_i)} \end{bmatrix}, \quad (6.74)$$

kjer smo interval $\lambda \in [0, 1]$ razdelili na N enakih delov, $\lambda_i = i/N$. Na sliki 6.9 je obravnavan isti primer kot na sliki 6.8, le da je namesto cenilke (6.53) uporabljena

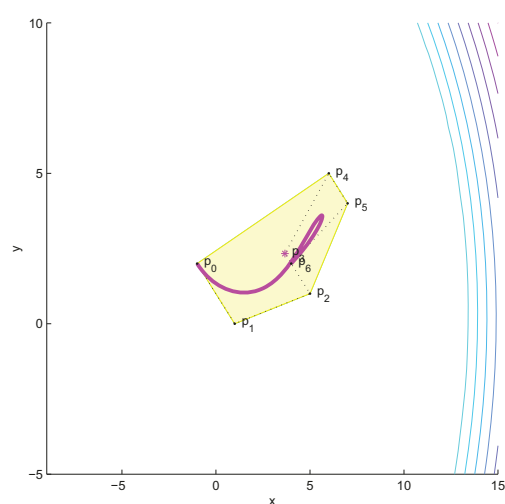


Slika 6.9: Optimalna kratka pot pri podanih zahtevah v robnih točkah in podanim časom trajanja trajektorije (optimizacija s prosto vmesno točko)..

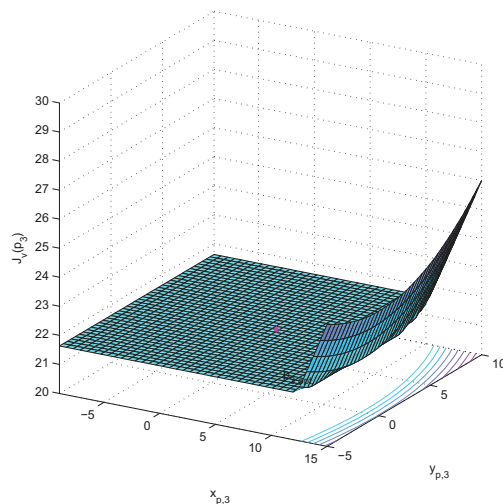
cenilka (6.54). Optimalna pot na sliki 6.9a ni najkrajša pot med robnima točkama, a je nekoliko bolj gladka.

Obravnavo optimalnih poti bi lahko nadaljevali še za kriterijske funkcije (6.55) do (6.59). Ker iščemo optimalno vrednost na zaprtem intervalu $\lambda \in [0, 1]$, je v določenim primerih možno, da imajo kriterijske funkcije v okolici optimalne vrednosti ravna dna. Kar je posledica iskanja maksimuma oz. minimuma v kriterijski funkciji, saj tako lahko kriterijska funkcija zasede maksimalno oz. minimalno vrednost ravno v robni točki. S tem optimizacijski problem postane slabo definiran. Zato je priporočljivo kombinirati kriterijske funkcije (6.55) do (6.59) z eno izmed kriterijskih funkcij (6.53) ali (6.54).

Pri optimizaciji kriterijske funkcije, ki ima več lokalnih minimumov, je rešitev odvisna od začetne ocene parametrov, kot je razvidno tudi s slik (6.11) do (6.14). Zanima nas, ali lahko pred začetkom ali med samo optimizacijo določimo podprostor $\Gamma \subset \Omega$, kjer se nahaja optimalna vrednost cenilke. Če to lahko storimo, lahko na ta način izboljšamo časovno učinkovitost optimizacije. Tak prostor je načeloma težko najti, zato na tem mestu problem nekoliko omilimo, tako da se vprašamo, ali obstaja določen del prostora $\Gamma_p \subset \Omega$, kjer je najbolj verjetno, da se nahaja minimum oz. lokalni minimumi kriterijske funkcije. Pri tem predpostavimo, da sta prostora Γ in Γ_p lahko v kakršnikoli medsebojni zvezi. Oblika prostora Γ_p (in

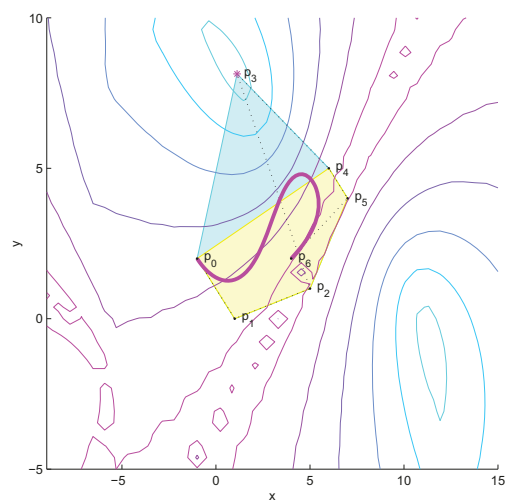


(a)

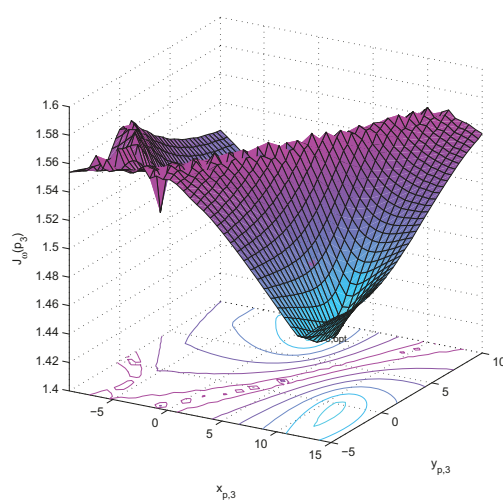


(b)

Slika 6.10: Optimalna maksimalna hitrost pri podanih zahtevah v robnih točkah in podanim časom trajanja trajektorije (optimizacija s prosto vmesno točko)..

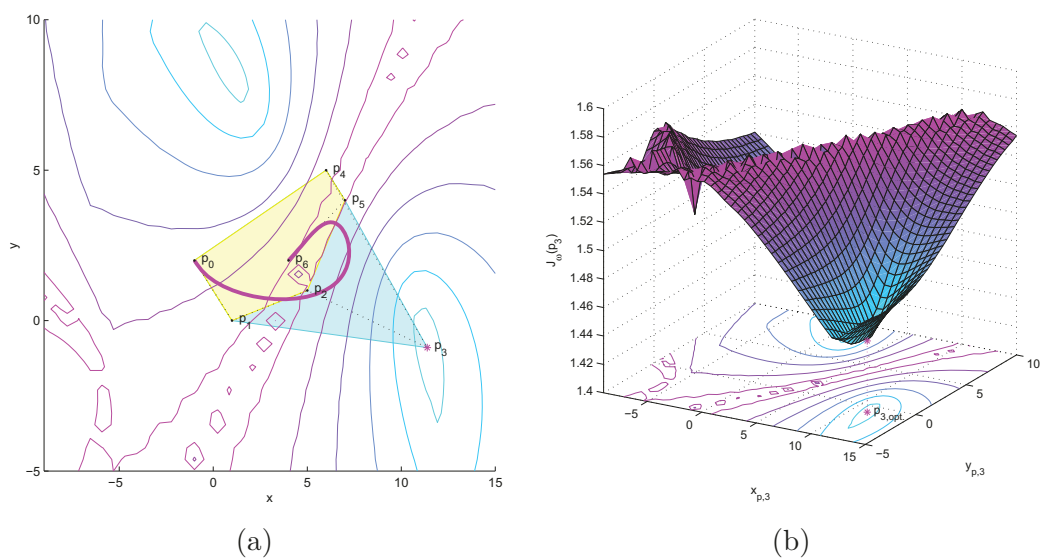


(a)

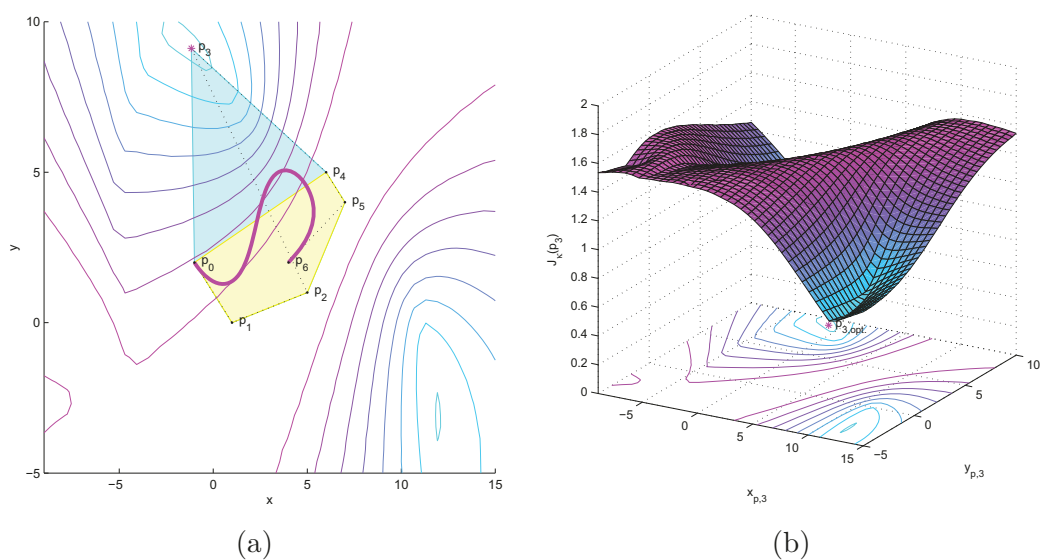


(b)

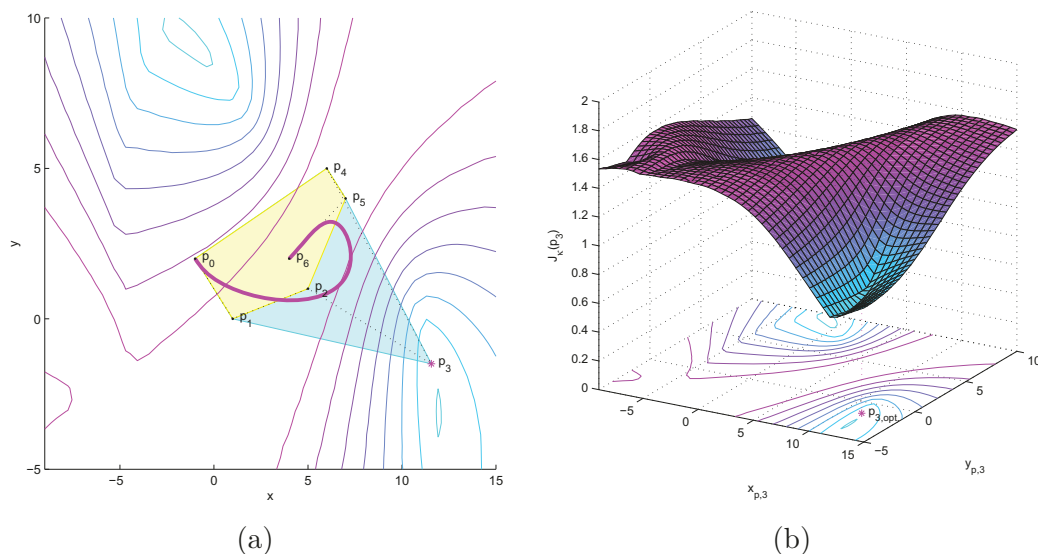
Slika 6.11: Optimalna maksimalna kotna hitrost pri podanih zahtevah v robnih točkah in podanim časom trajanja trajektorije (optimizacija s prosto vmesno točko)..



Slika 6.12: Optimalna maksimalna kotna hitrost pri podanih zahtevah v robnih točkah in podanim časom trajanja trajektorije (optimizacija s prosto vmesno točko).



Slika 6.13: Optimalna ukrivljenost pri podanih zahtevah v robnih točkah in podanim časom trajanja trajektorije (optimizacija s prosto vmesno točko).



Slika 6.14: Optimalna ukrivljenost pri podanih zahtevah v robnih točkah in podanim časom trajanja trajektorije (optimizacija s prosto vmesno točko).

tudi Γ) je seveda odvisna od kriterijske funkcije, zato obravnavamo problem za vsako kriterijsko funkcijo posebej.

Najprej se osredotočimo na primer, ko iščemo optimum, ki minimizira dolžino krivulje med robnima točkama. Napravimo razmislek. S premikanjem proste kontrolne točke spreminjamo obliko in tudi dolžino krivulje, ki povezuje mirujočo začetno in končno robno točko. Spomnimo se, da lahko vsako BB-krivuljo reda b enakovredno predstavimo z BB-krivuljo višjega reda $b + 1$ in da BB-krivulja vedno leži znotraj konveksne ovojnice, ki jo napenjajo kontrolne točke — ovojnica je napeta na Bézierjev poligon. Torej BB-krivulja (reda $b - 1$) brez dodatne vmesne proste kontrolne točke, leži znotraj pripadajoče ovojnice Bézierjevega poligona ∂_p . Z uvedbo dodatne vmesne proste kontrolne točke, lahko spremenimo obliko krivulje tako, da zapusti ovojnico prvotnega Bézierjevega poligona ∂_p — torej konveksno ovojnico, ki je napeta skozi vse, razen proste, kontrolne točke. Predstavljamo si lahko, da če spremenimo obliko krivulje na tak način, da le-ta zapusti konveksno območje, znotraj katerega se je nahajala na začetku ∂_p , da se v tem primeru najverjetneje podaljša dolžina krivulje, s tem pa se tudi poveča vrednost kriterijske funkcije, ki jo želimo minimizirati. Torej smo prišli do zaključka, da lahko optimalno vrednost kriterijske funkcije pričakujemo znotraj

oz. v okolici konveksne ovojnice ∂_p , ki jo napenjajo vse, razen proste, kontrolne točke. Ovojnica ∂_p je rob območja Γ_p .

Razmislimo, ali lahko podoben razmislek napravimo tudi v primeru, ko iščemo minimalno vrednost maksimalne tangencialne hitrosti. Ker je čas trajanja krivulje znan oz. določen, se s premikanjem proste kontrolne točke spreminja hitrostni profil krivulje. Zatorej lahko pri daljši krivulji pričakujemo večjo (maksimalno) tangencialno hitrost. Minimalno največjo hitrost krivulje bomo zato najverjetneje dosegli pri kratkih poteh, a ne nujno pri najkrajši poti. Vseeno lahko zopet omejimo iskanje optimalne vrednosti znotraj oz. v okolici območja Γ_p , saj gre le za najbolj verjetno območje. Do enakega rezultata pridemo tudi v primeru, ko napravimo razmislek za minimizacijo maksimalne kotne hitrosti in kotnega pospeška.

Ugotovili smo, da na podlagi takšnega razmisleka ne moremo vnaprej določiti področja, kjer leži optimalna rešitev, a najbolj verjetno se bo optimalna rešitev nahajala v omejeni okolici okrog kontrolnih točk. Najbolj pomembno dejstvo predhodnega razmisleka je, da ne glede na izbrano cenilko, dolžina krivulje pri optimalni vrednosti cenilke ni neskončno dolga, temveč je vedno omejena.

Optimizacija več cenilk

Običajno ne želimo zadostiti le enemu samemu pogoju, temveč želimo poiskati optimalno rešitev, ki zadosti več zahtevam hkrati. Pogost optimizacijski problem pri načrtovanju poti je iskanje najkrajše poti, ki hkrati zadosti tudi zahtevam (6.4). Minimizarati torej želimo kriterijsko funkcijo

$$\mathbf{p}_{\star, opt. s} = \arg \min_{\mathbf{p}_{\star}} J_s(\mathbf{p}_{\star}), \quad (6.75)$$

$$(6.76)$$

z omejitvami

$$|v(t)| - v_{max} \leq 0, \quad \forall t \in [t_a, t_b], \quad (6.77)$$

$$|\omega(t)| - \omega_{max} \leq 0, \quad \forall t \in [t_a, t_b], \quad (6.78)$$

$$|a(t)| - a_{max} \leq 0, \quad \forall t \in [t_a, t_b], \quad (6.79)$$

$$|\alpha(t)| - \alpha_{max} \leq 0, \quad \forall t \in [t_a, t_b]. \quad (6.80)$$

Omejitve (6.77) do (6.80) morajo biti izpolnjene v robnih točkah že pred samo optimizacijo. Optimizacijski problem lahko rešimo z optimizacijskimi metodami, ki so prirejene za reševanje optimizacijskih problemov z omejitvami. Lahko pa problem prevedemo tudi na optimizacijski problem brez omejitev, tako da omejitve implicitno uvedemo v kriterijsko funkcijo [108]. Optimizacijski problem lahko zatorej zapišemo v obliki:

$$\mathbf{p}_{\star, opt. s} = \arg \min_{\mathbf{p}_{\star}} J_s(\mathbf{p}_{\star}) + \quad (6.81)$$

$$+ c_v \max\{0, |v(t)| - v_{max}\} + \quad (6.82)$$

$$+ c_\omega \max\{0, |\omega(t)| - \omega_{max}\} + \quad (6.83)$$

$$+ c_a \max\{0, |a(t)| - a_{max}\} + \quad (6.84)$$

$$+ c_\alpha \max\{0, |\alpha(t)| - \alpha_{max}\}, \quad (6.85)$$

$$0 \leq t \leq \Lambda, \quad (6.86)$$

kjer uteži c_v , c_ω , c_a in c_α določajo vpliv oz. pomembnost posameznih omejitev. Tako zapisan optimizacijski problem lahko rešimo z metodami za reševanje optimizacijskih problemov brez omejitev. V primeru, ko imamo zlepek krivulj, moramo optimizacijski postopek ponoviti za vsako posamezno krivuljo, ki sestavlja celotni zlepek. V primeru zlepka, je možno definirati cenilko, ki optimizira celotni zlepek, a takšen način optimizacije ni računsko učinkovit.

6.2.4 Perspektivna transformacija

Ker je homografija nelinearna preslikava, v splošnem ne moremo preslikati BB-krivulje preko homografije v smislu kontrolnih točk. Vseeno lahko še vedno preslikamo točke na poljubni parametrični krivulji preko homografije, kot tudi lahko preslikamo njene odvode, če le-ti obstajajo. Zveznost in gladkost sta lastnosti krivulje, ki sta invariantni na preslikavo preko homografije. BB-krivulja se iz ravnine v prostoru preko homografije (3.4) preslika na sliko po pravilu:

$$P_{x\lambda}(\lambda) = \frac{\mathbf{h}_1^T W \mathbf{r}_\lambda(\lambda)}{\mathbf{h}_3^T W \mathbf{r}_\lambda(\lambda)} = \frac{\sum_{i=0}^b B_{i,b}(\lambda) \mathbf{h}_1^T W \mathbf{p}_i}{\sum_{i=0}^b B_{i,b}(\lambda) \mathbf{h}_3^T W \mathbf{p}_i}, \quad (6.87)$$

$$P_{y\lambda}(\lambda) = \frac{\mathbf{h}_2^T W \mathbf{r}_\lambda(\lambda)}{\mathbf{h}_3^T W \mathbf{r}_\lambda(\lambda)} = \frac{\sum_{i=0}^b B_{i,b}(\lambda) \mathbf{h}_2^T W \mathbf{p}_i}{\sum_{i=0}^b B_{i,b}(\lambda) \mathbf{h}_3^T W \mathbf{p}_i}. \quad (6.88)$$

Opozorimo, da smo tu izbrali homografijo med ravnino v prostoru in sliko, čeprav enako pravilo velja tudi za katerokoli drugo preslikavo preko homografije. Jasno je, da enačb (6.87) in (6.88), ki sta v obliki racionalne funkcije ne moremo zapisati v obliko polinoma (6.12). Sledi, da BB-krivulje ne moremo preslikati preko homografije v smislu kontrolnih točk.

Oglejmo si še, kako se preslikajo prvi odvodi preko homografije:

$${}^P\dot{x}_\lambda(\lambda) = \frac{{}^W\mathbf{r}_\lambda^T(\lambda)(\mathbf{h}_3\mathbf{h}_1^T - \mathbf{h}_1\mathbf{h}_3^T){}^W\dot{\mathbf{r}}_\lambda(\lambda)}{{}^W\mathbf{r}_\lambda^T(\lambda)\mathbf{h}_3\mathbf{h}_3^T{}^W\mathbf{r}_\lambda(\lambda)}. \quad (6.89)$$

$${}^P\dot{y}_\lambda(\lambda) = \frac{{}^W\mathbf{r}_\lambda^T(\lambda)(\mathbf{h}_3\mathbf{h}_2^T - \mathbf{h}_2\mathbf{h}_3^T){}^W\dot{\mathbf{r}}_\lambda(\lambda)}{{}^W\mathbf{r}_\lambda^T(\lambda)\mathbf{h}_3\mathbf{h}_3^T{}^W\mathbf{r}_\lambda(\lambda)}. \quad (6.90)$$

Drugi odvodi pa se preslikajo po pravilu:

$${}^P\ddot{x}_\lambda(\lambda) = \frac{{}^W\mathbf{r}_\lambda^T(\lambda)(\mathbf{h}_3\mathbf{h}_1^T - \mathbf{h}_1\mathbf{h}_3^T){}^W\ddot{\mathbf{r}}_\lambda(\lambda) - 2{}^W\dot{\mathbf{r}}_\lambda^T(\lambda)\mathbf{h}_3\mathbf{h}_3^T{}^W\dot{\mathbf{r}}_\lambda(\lambda){}^W\mathbf{r}_\lambda^T(\lambda)(\mathbf{h}_3\mathbf{h}_1^T - \mathbf{h}_1\mathbf{h}_3^T){}^W\dot{\mathbf{r}}_\lambda(\lambda)}{({}^W\mathbf{r}_\lambda^T(\lambda)\mathbf{h}_3\mathbf{h}_3^T{}^W\mathbf{r}_\lambda(\lambda))^2}, \quad (6.91)$$

$${}^P\ddot{y}_\lambda(\lambda) = \frac{{}^W\mathbf{r}_\lambda^T(\lambda)(\mathbf{h}_3\mathbf{h}_2^T - \mathbf{h}_2\mathbf{h}_3^T){}^W\ddot{\mathbf{r}}_\lambda(\lambda) - 2{}^W\dot{\mathbf{r}}_\lambda^T(\lambda)\mathbf{h}_3\mathbf{h}_3^T{}^W\dot{\mathbf{r}}_\lambda(\lambda){}^W\mathbf{r}_\lambda^T(\lambda)(\mathbf{h}_3\mathbf{h}_2^T - \mathbf{h}_2\mathbf{h}_3^T){}^W\dot{\mathbf{r}}_\lambda(\lambda)}{({}^W\mathbf{r}_\lambda^T(\lambda)\mathbf{h}_3\mathbf{h}_3^T{}^W\mathbf{r}_\lambda(\lambda))^2}. \quad (6.92)$$

6.2.5 Postopek načrtovanja poti

Oglejmo si sedaj celoten postopek načrtovanja poti po korakih. Na tem mestu se bomo ukvarjali s podajanjem poti preko slike, kar pomeni, da definiramo pot neposredno na sliki. Predstavljeni postopek je načeloma uporaben tako v primeru neposrednega kot tudi položajnega vodenja na osnovi slike. Pri načrtovanju predpostavimo, da končni zlepek spada v razred $\mathcal{C}^2$, kar pomeni, da so z določenimi vstopnimi lastnostmi v robno točko določene tudi izstopne lastnosti iz nje.

Korak 1: Izbira položajev robnih točk

V prvem koraku izberemo urejeno zaporedje položajev t. i. robnih točk, ki jih želimo obiskati oz. skozi katere želimo, da poteka krivulja. Vsaki dve zaporedni robni točki bomo v nadaljevanju povezali s BB-krivuljo. Ne glede na izbrani način regulacije vedno določimo točke neposredno na sliki, saj jih ob znani preslikavi med ravnino gibanja mobilnega sistema v prostoru lahko vedno enolično preslikamo s slike na ravnino v prostoru in obratno.

Korak 2: Izbira smernih vektorjev v robnih točkah

V drugem koraku vsaki robni točki, ki smo jo določili v prvem koraku, določimo smerni vektor, ki določa vhodno in izhodno smer tangente na krivuljo. Pri tej izbiri imamo več možnosti. Smerne vektorje v vseh robnih točkah lahko določimo ročno za vsako točko posebej, lahko pa uporabimo kakšno posebno pravilo za avtomatsko določitev smernih vektorjev v robnih točkah. Pogosta je izbira, ko za smerni vektor izberemo kar vektor, ki povezuje predhodno robno točko z naslednjo robno točko. Pri tem moramo ročno izbrati le smeri v začetni in končni robni točki, ali pa za le-ti dve točki uporabiti posebno pravilo. Na primer v začetni robni točki izberemo za smerni vektor kar vektor, ki povezuje začetno robno točko z naslednjo robno točko. Podobno storimo za končno robno točko. Orientacije v robnih točkah lahko preko preslikave (3.11) preslikamo s slike na ravnino v prostoru in obratno.

Korak 3: Določitev časovne funkcije

V tem koraku določimo časovno funkcijo oz. čase trajanja posameznih krivulj med robnimi točkami.

Korak 4: Določitev lastnosti v robnih točkah

V tem koraku določimo želene vstopne oz. izstopne hitrosti v robnih točkah. Ko določimo želene vstopne in izstopne hitrosti v robnih točkah, imamo v primeru, ko ni nobena od hitrosti v robnih točkah enaka nič, že določeni prvi in zadnji dve kontrolni točki, ki definirata posamezno krivuljo med sosednjima robnima točkama. Glede na to, na katere lastnosti v robnih točkah želimo vplivati, izberemo primeren red posamezne krivulje. Ko določimo vse lastnosti v robnih točkah oz. določimo položaje vseh kontrolnih točk, je s tem že določen zlepek krivulj, ki poteka skozi robne točke.

Korak 5: Optimizacija trajektorije

Sledi korak optimizacije trajektorije. V tem koraku lahko s podaljšanjem trajanja krivulje primerno preoblikujemo hitrostni profil krivulje, tako da zadostimo vsem hitrostnim omejitvam. Seveda pride s tem dejanjem do sprememb lastnosti, ki smo

jih določili v korakih 3 in 4. Optimizacijo krivulje lahko izvedemo tudi z vpeljavo dodatnih vmesnih točk na krivulji, ki omogočajo spreminjanje hitrostnega profila, brez spreminjanja hitrostnega profila v robnih točkah, a v tem primeru pride tudi do spremembe oblike krivulje. Če so kršene omejitve v robnih točkah s takšno obliko optimizacije seveda ne moremo zadostiti omejitvam. Tudi če niso kršene omejitve v robnih točkah, z optimizacijo s premikanjem vmesne kontrolne točke ni nujno, da obstaja rešitev, ki zadosti vsem zahtevam.

Korak 6: Časovno vzorčenje trajektorije

Ko je končna krivulja določena, izvedemo časovno vzorčenje trajektorije, glede na izbrano frekvenco delovanja diskretnega regulatorja za vodenje po trajektoriji. Vzorčeno trajektorijo preslikamo preko povezav (6.87), (6.88), (6.89), (6.90), (6.91) in (6.92) bodisi na sliko bodisi v svetovno ravnino.

7 Vodenje robotske roke

Robotska roka, ki je nameščena na mobilni sistem lahko močno razširi nabor možnih nalog, ki jih mobilni sistem, lahko opravlja. V tem poglavju je prikazana razširitev prikazane možnosti uporabe vodenja na osnovi slike za sledenje trajektoriji na mobilnih sistemih še za vodenje robotske roke po trajektoriji. Prikazana regulacijska struktura je zasnovana po principu dvoprostostnega vodenja. Obravnavamo primer postavitve, kjer imamo robotsko roko z več prostostnimi stopnjami in kamero, ki je vpeta v vrh robotske roke. Cilj regulacije je v tem primeru vodenje vrha robotske roke po trajektoriji. S pristopom direktnega vodenja na osnovi slike želimo regulirati šest prostostnih stopenj vrha robotske roke: položaj in orientacijo v 3D-prostoru. Regulacijski problem želimo rešiti neposredno na sliki. Problem je v tem primeru nekoliko drugačen kot pri vodenju mobilnega sistema po prostoru, saj predpostavimo, da referenčna trajektorija ni vezana na svetovni koordinatni sistem, temveč na lokalni koordinatni sistem kamere. Torej je tudi projekcija referenčne trajektorije na sliko časovno nespremenljiva. V kolikor je znan model kamere, lahko referenčno trajektorijo, ki je podana v koordinatnem sistemu kamere, preslikamo tudi na sliko.

7.1 Načrtovanje trajektorije

Predpostavimo, da je kinematični model robotske roke znan. Predpostavimo še, da za vrh robotskega mehanizma velja lastnost diferencialne gladkosti. Slednje nam omogoča zasnovo referenčne trajektorije iz primernokrat zvezno odvedljive parametrične krivulje. Kot smo že omenili, BB-krivulje niso omejene na dve dimenziji, temveč jih lahko definiramo v prostoru s poljubno dimenzijami — torej tudi v treh dimenzijah. Pri vodenju robotske roke obravnavamo problem vodenja

vrha robota. S kamero merimo in ocenjujemo pogrešek med trenutnim in želenim stanjem. Referenčno trajektorijo tako podamo med začetno in končno (želeno) lego v koordinatnem sistemu kamere. Regulirati želimo lego vrha robotske roke, kar pomeni, da moramo regulirati šest prostostnih stopenj (tri translacije in tri orientacije). Pri podajanju trajektorije moramo tako za vsak časovni trenutek podati tako zelen položaj kot želeno orientacijo kamere. Referenčni položaj in orientacijo lahko podamo neodvisno ali odvisno, pri čemer orientacijo določimo na podlagi odvodov referenčne poti.

Eno takšnih predstavitev omogoča Frenet-Serretjev koordinatni sistem, ki opisuje kinematični model gibanja delca po zvezno odvedljivi krivulji v 3D-prostoru. Frenet-Serrejev koordinatni sistem sestavljajo enotski vektorji $\mathbf{e}_T$, $\mathbf{e}_N$ in $\mathbf{e}_B$, ki jih lahko določimo iz dvakrat zvezno odvedljive parametrične 3D-krivulje $\mathbf{r}(s)$. Parameter s definiramo kot dolžino krivulje:

$$s(t) = \int_0^t \left\| \frac{d\mathbf{r}(\tau)}{d\tau} \right\| d\tau. \quad (7.1)$$

V primeru, da izraz pod integralom (7.1) ni nikoli enak nič, je dolžina poti $s(t)$ monotonno naraščajoča funkcija, kar omogoča zapis krivulje $\mathbf{r}(t)$ kot funkcijo dolžine krivulje $\mathbf{r}(s) = \mathbf{r}(t(s))$. Tako Frenet-Serrejev koordinatni sistem definirajo naslednje enačbe:

$$\mathbf{e}_T = \frac{\partial \mathbf{r}}{\partial s}, \quad (7.2)$$

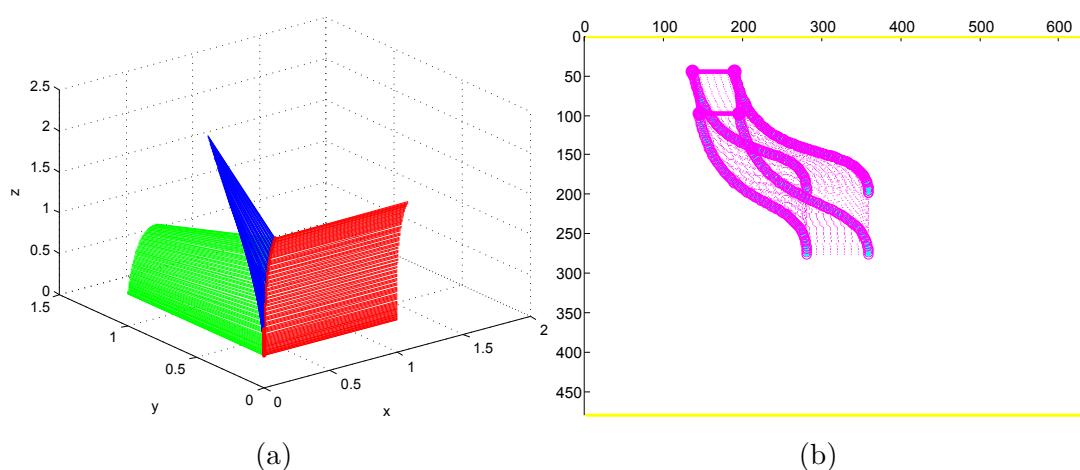
$$\mathbf{e}_N = \frac{\frac{\partial \mathbf{e}_T}{\partial s}}{\left\| \frac{\partial \mathbf{e}_T}{\partial s} \right\|}, \quad (7.3)$$

$$\mathbf{e}_B = \mathbf{e}_T \times \mathbf{e}_N. \quad (7.4)$$

Orientacijo lahko v 3D-prostoru zapišemo na različne načine (z eno izmed različic Eulerjevih kotov, z rotacijsko matriko, s kvaternioni). Določeni zapisi lahko v določenih pogojih postanejo singularni, zato je še posebej primeren zapis orientacije s kvaternioni, ki te pomanjkljivosti nimajo. Kot smo omenili, lahko referenčni položaj izberemo neodvisno od referenčne orientacije. Referenčno orientacijo lahko podamo za vsak časovni trenutek posebej, lahko pa referenčne orientacije v vsakem trenutku določim s pomočjo interpolacije med začetno in končno želeno orientacijo. V primeru predstavitve orientacije z Eulerjevimi koti

enostavne linearne interpolacije posameznih kotov ne prinašajo lepega prehoda iz začetne v končno orientacijo. V primeru predstavitve orientacije s kvaternioni lahko izvedemo interpolacijo z algoritmom SLERP (angl. „*Spherical Linear interpolation*”), ki zagotovi vrtenje začetne orientacije s konstantno kotno hitrostjo okoli ene same osi v končno orientacijo [173].

V poglavju 6 so predstavljene nekatere posebne lastnosti Bernstein-Bézierjevih krivulj v 2D-prostoru, ki jih lahko koristno uporabimo pri načrtovanju referenčne trajektorije. V poglavju 6 je izpostavljeno, kako z različnimi postavitvami kontrolnih točk vplivamo na lastnosti krivulje v robnih točkah. Podobno obravnavo je možno razširiti tudi na prostorske krivulje. Očitno je, da robne kontrolne točke predstavljajo izvir oz. ponor krivulje. Opazimo lahko tudi, da druga kontrolna oz. predzadnja kontrolna točka vpliva na izstopni oz. vstopni kot in začetno oz. končno tangencialno hitrost v robni točki. Opisana opažanja bi lahko opisali tudi z enačbami, kot je to za dvodimenzionalni primer storjeno v poglavju 6. Na sliki 7.1a je prikazan primer referenčne trajektorije v 3D-prostoru in prikaz trajektorij štirih referenčnih točk na sliki.



Slika 7.1: Primer referenčne trajektorije kamere v (a) 3D-prostoru in prikaz trajektorij štirih referenčnih točk na sliki.

7.2 Regulator za sledenje trajektoriji

Predpostavimo, da imamo opravka z robotskim mehanizmom, ki ima šest ($D = 6$) rotacijskih sklepov. Zasuke posameznih sklepov opišemo z vektorjem zasukov $\mathbf{g}(t) = [\varphi_i(t)]_{i=1,2,\dots,D}$. Ob znanem kinematičnem modelu robotskega mehanizma in znanih zasukih posameznih sklepov, lahko zapišemo lego vrha robotskega mehanizma (na vrh robota je pritrjen koordinatni sistem R) glede na svetovni koordinatni sistem W , ki jo zapišemo s translacijo v obliki vektorja ${}^W\mathbf{t}_R(t) \in \mathbb{R}^3$ in orientacijo v obliki kvaterniona ${}^W\hat{\mathbf{q}}_R(t) \in \mathbb{H}$:

$${}^W\mathbf{t}_R(t) = {}^W\mathbf{t}_R(\mathbf{g}(t)), \quad {}^W\hat{\mathbf{q}}_R(t) = {}^W\hat{\mathbf{q}}_R(\mathbf{g}(t)). \quad (7.5)$$

Hitrosti v posameznih sklepih zapišemo z vektorjem hitrosti $\dot{\mathbf{g}}(t) = [\dot{\varphi}_i(t)]_{i=1,2,\dots,D}$. Hitrost vrha robotskega mehanizma zapišemo z vektorjem hitrosti vijačnega premika ${}^R\mathbf{u}^T(t) = [{}^R\mathbf{v}^T(t), {}^R\boldsymbol{\omega}^T(t)] \in \mathbb{R}^6$, kjer je ${}^R\mathbf{v}^T(t) = [{}^Rv_x, {}^Rv_y, {}^Rv_z]$ vektor translatorskih hitrosti in ${}^R\boldsymbol{\omega}^T(t) = [{}^R\omega_x, {}^R\omega_y, {}^R\omega_z]$ vektor kotnih hitrosti. Iz kinematičnega modela mobilnega sistema lahko s pomočjo Taylorjeve vrste določimo povezavo:

$${}^R\mathbf{u}(t) = \mathbf{J}(\mathbf{g}(t))\dot{\mathbf{g}}(t), \quad (7.6)$$

kjer je Jacobijeva matrika $\mathbf{J}(\mathbf{g}(t)) \in \mathbb{R}^6 \times \mathbb{R}^D$ časovno spremenljiva in odvisna od trenutne postavitve robotskega mehanizma oz. od notranjih zasukov sklepov. Hitrosti v sklepih $\dot{\mathbf{g}}(t)$ določimo na podlagi ocenjene inverzne Jacobijeve matrike:

$$\dot{\mathbf{g}}(t) = \widehat{\mathbf{J}^{-1}}(\mathbf{g}(t)){}^R\mathbf{u}(t). \quad (7.7)$$

Inverznega kinematičnega modela običajno ne moremo izraziti v analitični obliki, zato rešitev običajno poiščemo s pomočjo numeričnih metod. Zelo pogosta je uporaba Newton-Raphsonove iterativne numerične metode, pri čemer se pri izračunu rešitve uporablja Jacobijeva matrika $\mathbf{J}(\mathbf{g}(t))$. Izračun inverzne kinematike je zahtevna naloga, saj imajo robotski mehanizmi običajno določene singularne postavitve, kjer problem ni rešljiv, ali pa obstaja pri določenih postavitvah več rešitev. Pri robotskih mehanizmih z veliko prostostnimi stopnjami imamo opravka z odvečnimi prostostnimi stopnjami, zato se pri takšnih sistemih vedno srečujemo s problemom več rešitev. Odvečne prostostne stopnje lahko koristno uporabimo za izogibanje oviram brez vpliva na lego vrha robotskega mehanizma.

Obravnavamo postavitev, kjer je kamera vpeta v sam vrh robotskega mehanizma. A kamere običajno ne moremo namestiti tako, da bi bil koordinatni sistem kamere poravnani s koordinatnim sistemom vrha robota. Predpostavimo, da lego kamere glede na vrh robotskega mehanizma podajata časovno nespremenljivi vektor translacije ${}^R\mathbf{t}_C \in \mathbb{R}^3$ in kvaternion orientacije ${}^Rq_C \in \mathbb{H}$. Vektor hitrosti vijačnega premika vrha robotskega mehanizma ${}^R\mathbf{u}(t)$ lahko tako izrazimo z vektorjem hitrosti vijačnega premika kamere ${}^C\mathbf{u}(t)$ s povezavo:

$${}^R\mathbf{u}(t) = {}^R\mathbf{L}_C {}^C\mathbf{u}(t), \quad {}^R\mathbf{L}_C = \begin{bmatrix} {}^R\mathbf{R}_C & [{}^C\mathbf{t}_R]_{\times} {}^R\mathbf{R}_C \\ \mathbf{0} & {}^R\mathbf{R}_C \end{bmatrix}, \quad (7.8)$$

kjer matrika ${}^R\mathbf{R}_C \in \text{SO}(3)$ predstavlja orientacijo Rq_C zapisano v obliki rotacijske matrike.

Preden nadaljujemo nekoliko poenostavimo notacijo, tako da opustimo pisanje indeksov C in P : ${}^C\mathbf{u}^T(t) = \mathbf{u}^T(t) = [v_x, v_y, v_z, \omega_x, \omega_y, \omega_z]$, ${}^C\mathbf{p} = [{}^Cx, {}^Cy, {}^Cz] = [X, Y, Z]$ in ${}^P\mathbf{p} = [{}^Px, {}^Py] = [x, y]$. Predpostavimo, da na sliki opazujemo značilke, ki jih opišemo z vektorjem značilk $\mathbf{s}(t) \in \mathbb{R}^K$, ki je dolžine K . Hitrost spreminjanja vektorja značilk na sliki podamo z modelom:

$$\dot{\mathbf{s}}(t) = \mathbf{f}(\mathbf{s}(t), \mathbf{u}(t)). \quad (7.9)$$

Izdelati želimo regulator, ki bo zagotovil sledenje vektorja značilk $\mathbf{s}(t)$ referenčni časovni funkciji $\mathbf{s}_r(t)$ na sliki. Referenčno trajektorijo zasnujemo v koordinatnem sistemu kamere (poglavje 7.1) in glede na znan model kamere določimo časovni potek vektorja značilk $\mathbf{s}_r(t)$. Regulator za sledenje referenčni trajektoriji lahko zasnujemo po principu dvoprostostne regulacije. Na podlagi referenčne trajektorije lahko izračunamo odprtozančno krmilno akcijo $\mathbf{u}_F(t)$. Ker smo referenčno trajektorijo podali v koordinatnem sistemu kamere, moramo pri izračunu inverzne transformacije poznati še absolutni položaj začetne / vmesne / končne referenčne lege glede na svetovni koordinatni sistem robotskega mehanizma. Regulacijski pogrešek odpravimo tako, da na sliki opazujemo določene značilke, pri čimer imamo na voljo več možnosti. Predpostavimo, da imamo na voljo sistem za strojno sledenje N točkam na sliki, torej je dolžina vektorja značilk $K = 2N$. Predpostavimo, da imamo opravka z enostavnim modelom kamere (3.2), kjer je $\gamma = 0$ in $\alpha = \beta = f$, kjer je f goriščna razdalja kamere. Z odvajanjem točkovnih značilk pridemo do nelinearnega modela značilk (7.9), ki ga lahko lineariziramo

okoli referenčne trajektorije:

$$\dot{\mathbf{s}}(t) = \mathbf{A}(t)\mathbf{s}(t) + \mathbf{B}(t)\mathbf{u}(t), \quad (7.10)$$

kjer je $\mathbf{A}^T = \text{diag}([\mathbf{A}_1^T \ \mathbf{A}_2^T \ \dots \ \mathbf{A}_N^T])$:

$$\mathbf{A}_i(t) = \begin{bmatrix} \frac{v_{z,r}(t)}{Z_{i,r}(t)} + \omega_{x,r}(t)y_{i,r}(t)f + \frac{2\omega_{y,r}(t)x_{i,r}(t)}{f} & \frac{\omega_{x,r}(t)x_{i,r}(t)}{f} - \omega_{z,r}(t) \\ \omega_{y,r}(t)y_{i,r}(t) + \omega_{z,r}(t) & \frac{v_{z,r}(t)}{Z_{i,r}(t)} + \frac{2\omega_{x,r}(t)y_{i,r}(t)}{f} + \omega_{y,r}(t)x_{i,r}(t) \end{bmatrix} \quad (7.11)$$

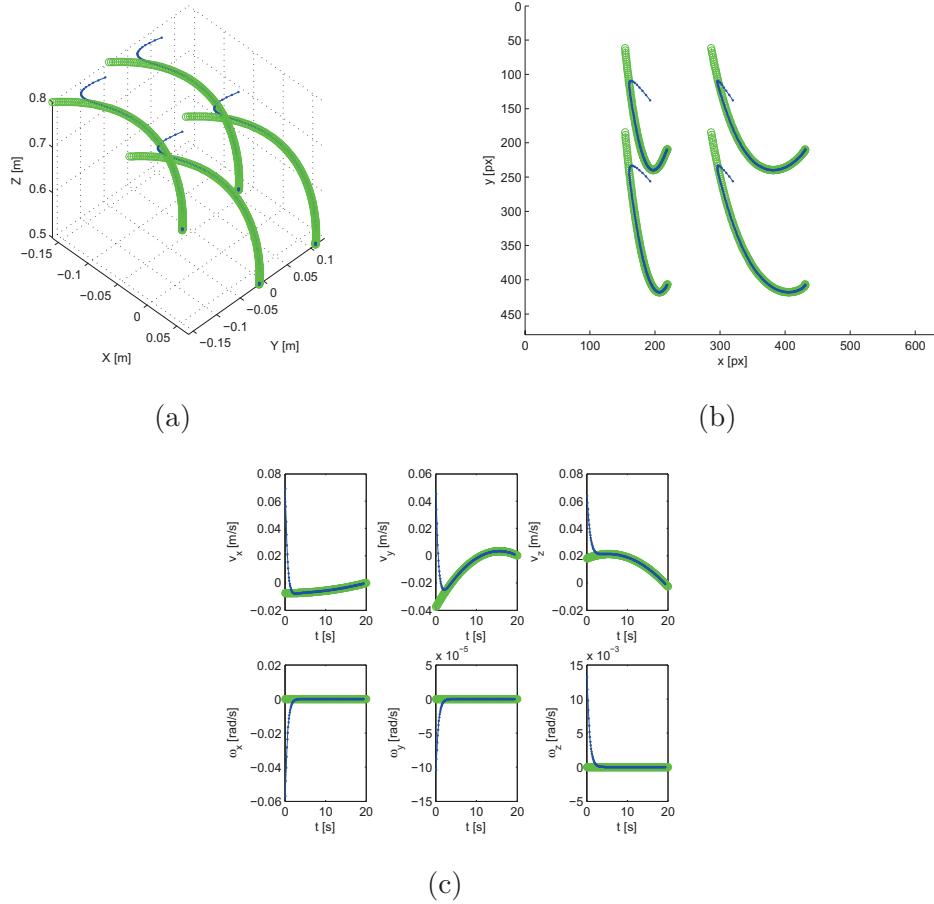
in $\mathbf{B}^T = [\mathbf{B}_1^T \ \mathbf{B}_2^T \ \dots \ \mathbf{B}_N^T]$:

$$\mathbf{B}_i(t) = \begin{bmatrix} \frac{f}{Z_{i,r}(t)} & 0 & \frac{x_{i,r}(t)}{Z_{i,r}(t)} & \frac{x_{i,r}(t)y_{i,r}(t)}{f} & f \left(1 + \left(\frac{x_{i,r}(t)}{f} \right)^2 \right) & -y_{i,r}(t) \\ 0 & -\frac{f}{Z_{i,r}(t)} & \frac{y_{i,r}(t)}{Z_{i,r}(t)} & f \left(1 + \left(\frac{y_{i,r}(t)}{f} \right)^2 \right) & x_{i,r}(t)y_{i,r}(t) & x_{i,r}(t) \end{bmatrix}. \quad (7.12)$$

V enačbah (7.11) in (7.12) smo referenčne vrednosti označili z oznako $\cdot_r$. Pri izpeljavi smo upoštevali relacijo: ${}^C\dot{\mathbf{p}} = -\mathbf{v} - [\boldsymbol{\omega}]_{\times} {}^C\mathbf{p}$. Ob upoštevanju, da model (7.10) velja tudi za referenčni signal $\mathbf{s}_r(t)$ in $\mathbf{e}(t) = \mathbf{s}_r(t) - \mathbf{s}(t)$ ter $\mathbf{u}(t) = \mathbf{u}_F(t) - \mathbf{u}_B(t)$, lahko izpeljemo naslednji model pogreška:

$$\dot{\mathbf{e}}(t) = \mathbf{A}(t)\mathbf{e}(t) + \mathbf{B}(t)\mathbf{u}_B(t). \quad (7.13)$$

Model pogreška (7.13) lahko uporabimo v izvedbi prediktivnega regulatorja (poglavje 5.4). Za izvedo vodenje vseh šestih prostostnih stopenj moramo slediti najmanj $N = 3$ ($K = 6$) točkam na sliki. Na sliki 7.2 so prikazani simulacijski rezultati vodenja vrha robotske roke po trajektoriji s prediktivnim regulatorjem, pri čemer sledimo $N = 4$ točkam na sliki. Izbrani so bili naslednji parametri prediktivnega regulatorja: $T = 0,1 \text{ s}$, $h = 5$, $\mathbf{Q} = \text{diag}(5)$, $\mathbf{R} = \text{diag}(5000)$, $\mathbf{A}_r = \text{diag}(0.85)$. Regulator je zmožen odpraviti pogrešek in zagotoviti sledenje referenčni trajektoriji. Sledilno delovanje regulatorja je dobro le pri majhnem orientacijskem pogrešku, saj smo uporabili pristop direktnega vodenja na osnovi slike, kjer smo pogrešek določili na podlagi razlike točk. Za primerno delovanje regulacijske zanke je ključna izbira značilnk [2, 4].



Slika 7.2: Vodenje kamere po trajektoriji s prediktivnim regulatorjem (sledenje štirim točkam). Referenčna in dejanska trajektorija v (a) 3D-prostoru kamere in (b) na sliki ter (c) poteki tangencialnih in kotnih hitrosti kamere.

8 Sklep

V delu je bilo predstavljeno načrtovanje vodenja mobilnega sistema in robotske roke s pristopom vodenja na osnovi slike (VOS). Pri tem smo se osredotočali na obravnavo nekanoničnih postavitev kamere, ki so praktično uporabne, a zaradi nelinearnosti perspektivne transformacije niso enostavno rešljive. Na podlagi obravnave lastnosti perspektivne projekcije ravnine smo zasnovali tudi metodo za kalibracijo kamere, ki ne zahteva posebnega prirejanja okolja le za namen kalibracije. Pri načrtovanju sistemov vodenja smo upoštevali nekatere posebne lastnosti preslikave ravnine preko perspektivne projekcije. S pomočjo takšne obravnave smo uspeli rešiti določene probleme vodenja na osnovi slike, kot je vodenje mobilnega sistema po trajektoriji. Algoritem vodenja po trajektoriji smo izvedli v diskretni obliki z dvoprostostno regulacijsko shemo. Odprtozančni krmilni signal smo izračunali iz referenčne trajektorije ob upoštevanju znanega kinematičnega modela sistema. V idealnem primeru naj bi tako referenčni krmilni signal zagotovil idealno sledenje referenčni trajektoriji brez pogreška. Za odpravo regulacijskega pogreška smo krmilnemu signalu v direktni veji dodali še regulirni signal v povratni zanki. Meritev pogreška smo opravili na podlagi strojne obdelave slik, pri čemer smo uporabili pristop direktnega vodenja na osnovi slike. Povratnozančni regulator smo izvedli po pristopu prediktivnega vodenja z upoštevanjem modela pogreška. Model pogreška smo podali kot časovno spremenljiv diskretni sistem v prostoru stanj. Uporabili smo različne modele pogreška. Med drugim smo predlagali model pogreška, ki naj bi omogočal izvedbo stabilnega regulatorja za sledenje referenci tudi v primeru, ko referenčna hitrost doseže ničelno tangencialno hitrost. Čeprav pri ničelni tangencialni hitrosti predlagana regulacijska struktura izniči vpeljeni pomožni pogrešek, pa v splošnem ne izniči tudi dejanskega pogreška. Predstavili smo postopek načrtovanja referenčne trajektorije, ki temelji na uporabi Bernstein-Bézierjevih krivulj. Pri tem smo za dvodimenzionalni

prostor obravnavali določene posebne primere, ki nastopijo pri ničelnem, prvem ali višjem odvodu tangencialne smeri krivulje (tangencialna hitrost, pospešek in druge lastnosti trajektorije). Pokazali smo, da lahko pri načrtovanju trajektorije z Bernstein-Bézierjevimi krivuljami, določimo referenčni regulirni signal za vodenje mobilnega sistema po ravnini tudi za ničelno tangencialno referenčno hitrost — ugotovitev velja tudi za splošne parametrične krivulje, ki so primernokrat zvezno odvedljive ali gladke. Prikazali smo tudi, kako posamezne kontrolne točke Bernstein-Bézierjevih krivulj vplivajo na lastnosti krivulje v začetnih in končnih točkah. Obravnavo smo napravili za dvodimenzionalni primer do mere, ki je praktično uporabna. Predstavljeno obravnavo je možno nadaljevati in uporabiti tudi za višjedimenzionalne prostore. Obravnavali smo tudi različne možnosti optimizacije trajektorije, kjer smo se osredotočali na iskanje analitično rešljivih optimizacijskih problemov. Nakazali smo, kako lahko predstavljeno metodologijo vodenja avtonomnega mobilnega sistema po trajektoriji uporabimo tudi za izvedbo vodenja vrha robotske roke po trajektoriji. Pri tem smo določene rešitve primerno razširili za obravnavo v 3D-prostoru.

9 Doprinos k znanosti

- Izdelava simulacijskega orodja, matematičnih modelov in nabora večnivojskih modularnih algoritmov za reševanje problema vodenja na osnovi slike.
- Izdelava metode za kalibracijo kamere, ki temelji na vrtenju točke okoli osi s konstantno hitrostjo. Metoda je bila zasnovana za uporabo v aplikacijah vodenja na osnovi slike.
- Izvedba prediktivnega regulatorja na osnovi modela pogreška za vodenje po trajektoriji, ki omogoča tudi zaustavitev na trajektoriji.
- Predstavitev načina načrtovanja trajektorije na podlagi zlepkov parametričnih krivulj z obravnavo posebnih primerov v robnih točkah.
- Predstavitev lokalizacije mobilnega sistema na osnovi slike z uporabo geometrijskih značilk po pristopu vizualne odometrije.
- Zasnova regulatorja za vodenje vrha robotske roke na osnovi slike po trajektoriji.

Dodatek

A Preslikave med koordinatnimi sistemi

Lego kamere v prostoru lahko opišemo s homogeno transformacijsko matriko ${}^W\mathbf{T}_C \in \text{SE}(3)$, ki podaja lego koordinatnega sistema kamere (oznaka C) glede na osnovni koordinatni sistem (oznaka W). Ob upoštevanju, da obratno transformacijo predstavlja inverzna matrika ${}^C\mathbf{T}_W = {}^W\mathbf{T}_C^{-1}$ lahko zapišemo preslikavo poljubne točke iz osnovnega koordinatnega sistema ${}^W\mathbf{p}$ v koordinatni sistem kamere ${}^C\mathbf{p}$:

$${}^C\mathbf{p} = {}^C\mathbf{T}_W {}^W\mathbf{p}. \quad (\text{A.1})$$

Ker predstavlja homogena transformacija ${}^W\mathbf{T}_C$ le kombinacijo premika in rotacije, ne pa tudi skaliranja, jo lahko predstavimo v obliki:

$${}^W\mathbf{T}_C = \begin{bmatrix} {}^W\mathbf{R}_C & {}^W\mathbf{t}_C \\ \mathbf{0}^T & 1 \end{bmatrix}, \quad (\text{A.2})$$

kjer vektor ${}^W\mathbf{t}_C \in \mathbb{R}^3$ podaja položaj koordinatnega izhodišča kamere C glede na osnovni koordinatni sistem W , medtem ko matrika ${}^W\mathbf{R}_C \in \text{SO}(3)$ opisuje rotacijo iz osnovnega v koordinatni sistem kamere. Lastnost ortogonalnosti rotacijske matrike ${}^W\mathbf{R}_C$ vodi do enostavnega izračuna inverzne matrike (A.2):

$${}^W\mathbf{T}_C^{-1} = {}^W\mathbf{T}_C^T = {}^C\mathbf{T}_W = \begin{bmatrix} {}^W\mathbf{R}_C^T & -{}^W\mathbf{R}_C^T {}^W\mathbf{t}_C \\ \mathbf{0}^T & 1 \end{bmatrix}. \quad (\text{A.3})$$

B Geometrijski model kamere

Kadar preslikavamo točke iz prostora z N dimenzijami v prostor z $M < N$ dimenzijami, govorimo o t. i. *projekcijah*. Če je gorišče preslikave pomaknjeno v neskončnost, govorimo o paralelni projekciji, sicer pa o perspektivni projekciji. Perspektivna projekcija je nelinearna transformacija, medtem ko je paralelna projekcija linearna. Ob določenih nastavitvah kamere in njeni postavitvi lahko perspektivno transformacijo dovolj dobro predstavimo z ustreznim linearnim približkom [4]; kar med drugim omogoča enostavno kalibracijo kamere.

V nadaljevanju je predstavljen zapis modela tanke leče z goriščno razdaljo f . Koordinatno izhodišče kamere postavimo v optično središče kamere. Os Cz leži na optični osi in kaže v smeri pogleda, os Cx kaže v desno in os Cy navzdol kot je prikazano na sliki B.1 — smeri osi so izbrane tako, da je kamera predstavljena v desnoučnem koordinatnem sistemu. Za namen izpeljave vpeljemo dodatni koordinatni sistem D , ki sovpada¹ s koordinatnim sistemom C . Čisto perspektivno transformacijo podaja izraz:

$$^D\mathbf{p} = \frac{1}{\lambda} {}^D\mathbf{T}_C {}^C\mathbf{p}, \quad (\text{B.1})$$

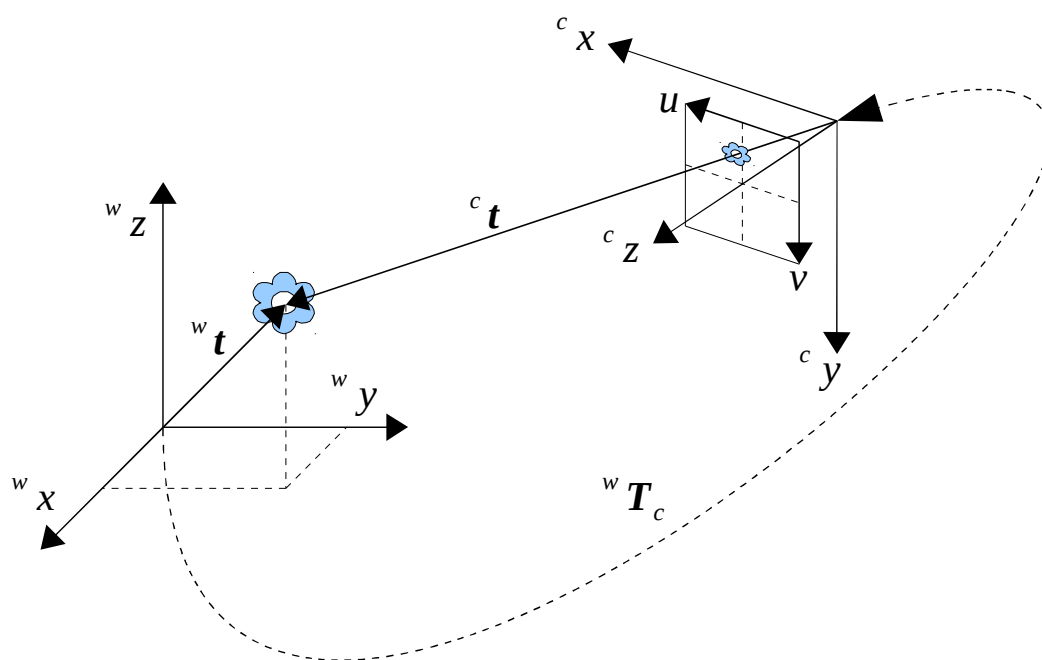
kjer sta $^D\mathbf{T}_C$ in λ podana z izrazoma:

$$^D\mathbf{T}_C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{f} & 1 \end{bmatrix}, \quad \frac{1}{\lambda} = \frac{f}{f - c_z}. \quad (\text{B.2})$$

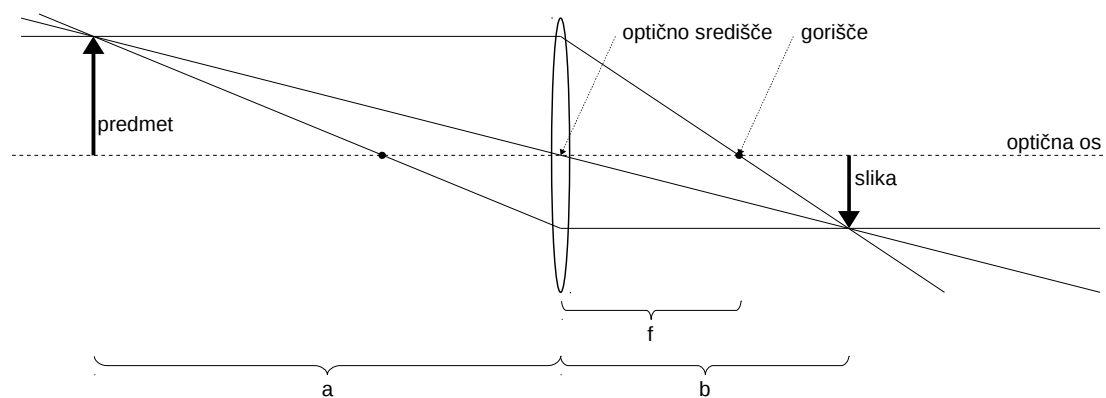
Enačbo (B.1) lahko zapišemo tudi nekoliko preprosteje:

$$^D\mathbf{p} = \frac{1}{\lambda} {}^C\mathbf{p} \quad \text{oz. kar} \quad {}^D\mathbf{p} \simeq {}^C\mathbf{p}. \quad (\text{B.3})$$

¹Koordinatna sistema sovpadata, če imata skupno koordinatno izhodišče in enaki orientaciji.



Slika B.1: Predstavitev lege kamere in slike v prostoru.



Slika B.2: Model tanke leče.

saj je podmatrika matrike ${}^D\mathbf{T}_C$ brez zadnjega stolpca in zadnje vrstice enotina matrika. Položaj člena $-\frac{1}{f}$ v tretjem stolpcu matrike ${}^D\mathbf{T}_C$ določa, da gre za perspektivno transformacijo vzdolž osi Cz . Glede na izbrano postavitev koordinatnih sistemov in enačbe (B.1) in (B.2), nastane pri ${}^Cz > f$ pomanjšana in obrnjena slika na zaslonu za lečo. Zaradi lažje predstavljenosti, pogosto obrnemo predznak elementov v zadnji vrstici matrike ${}^D\mathbf{T}_C$ in tudi predznak faktorja λ , saj si v tem primeru lahko predstavljamo, da za ${}^Cz > f$ nastane pomanjšana slika pred lečo in ni obrnjena. Zadnja vrstica v matriki ${}^D\mathbf{T}_C$ opisuje t. i. *ravnino v neskončnosti*, za katero velja, da točke s te ravnine tvorijo sliko v neskončnosti. Pri izbiri parametrov (B.2), je to ravnina ${}^Cz = f$. Torej za točke, ki se nahajajo na razdalji f od leče, slika ne nastane. Za točke, ki so od leče oddaljene manj kot je goriščna razdalja leče (${}^Cz < f$), nastane povečana slika pred lečo. Mesto nastanka ostre slike podaja izraz:

$${}^Dz = \frac{f {}^Cz}{f - {}^Cz}, \quad (\text{B.4})$$

ki je pravzaprav enačba tanke leče, ki jo običajno srečamo v naslednji obliki:

$$-\frac{1}{{}^Cz} + \frac{1}{{}^Dz} = -\frac{1}{f}. \quad (\text{B.5})$$

Pri velikih razdaljah, ${}^Cz \gg f$, lahko člen f v imenovalcu (B.4) zanemarimo, kar nekoliko poenostavi model (B.2):

$${}^D\mathbf{T}_C \approx \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\frac{1}{f} & 0 \end{bmatrix}, \quad \frac{1}{\lambda} \approx -\frac{f}{{}^Cz}. \quad (\text{B.6})$$

Opazimo, da v tem primeru nastane slika ravno v gorišču za lečo (${}^Dz \approx -f$). V tem primeru se *ravnina v neskončnosti* pomakne ravno v optično središče leče (${}^Cz = 0$). Z modela tanke leče smo prešli na model *kamere na luknjico*, ki je eden izmed najbolj pogosto uporabljenih modelov pri načrtovanju VOS. V nadaljevanju bomo zato upoštevali model (B.6), čeprav je mogoče nekatere relacije izpeljati tudi za model tanke leče oz. celo splošneje.

Na mesto nastanka slike postavimo zaslon s tipalom za zajem slike. Na slikovni zaslon postavimo koordinatni sistem slike (oznaka P), ki ima izhodišče običajno

v zgornjem levem kotu slike, enote pa so podane glede na velikost slikovnega elementa. Celotna preslikava se torej glasi:

$${}^P\mathbf{p} = {}^P\mathbf{T}_D {}^D\mathbf{p} = {}^P\mathbf{T}_D \left(\frac{1}{\lambda} {}^D\mathbf{T}_C \right) {}^C\mathbf{p} = \frac{1}{\lambda} {}^P\mathbf{T}_C {}^C\mathbf{p}, \quad (\text{B.7})$$

kjer je:

$${}^P\mathbf{T}_D = \begin{bmatrix} k_u & k_{uv} & 0 & u_c \\ 0 & k_v & 0 & v_c \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (\text{B.8})$$

V sistemu ${}^P\mathbf{T}_D {}^D\mathbf{T}_C$ matrika (B.8) pobriše tretjo vrstico matrike ${}^D\mathbf{T}_C$, zato lahko za ${}^P\mathbf{T}_C \in \mathbb{R}^3 \times \mathbb{R}^4$ zapišemo:

$${}^P\mathbf{T}_C = \begin{bmatrix} k_u & k_{uv} & u_c \\ 0 & k_v & v_c \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{1}{f} & 0 \end{bmatrix} = [\mathbf{S} \quad \mathbf{0}], \quad (\text{B.9})$$

kar poenostavi zapis (B.7):

$${}^P\mathbf{p} = \frac{1}{\lambda} \mathbf{S} {}^C\mathbf{p} \quad \text{oz. še preprosteje} \quad {}^P\mathbf{p} \simeq \mathbf{S} {}^C\mathbf{p}. \quad (\text{B.10})$$

Matrika $\mathbf{S} \in \mathbb{R}^3 \times \mathbb{R}^3$ opisuje *notranji model kamere*:

$$\mathbf{S} = \begin{bmatrix} k_u & k_{uv} & -\frac{u_c}{f} \\ 0 & k_v & -\frac{v_c}{f} \\ 0 & 0 & -\frac{1}{f} \end{bmatrix}. \quad (\text{B.11})$$

Parametri $k_u, k_v, k_{uv}, u_c, v_c$ in f v matriki $\mathbf{S}$ so t. i. *notranji parametri kamere*. Preslikava točke ${}^W\mathbf{p} \in \mathbb{R}^4$ v poljubnem koordinatnem sistemu W v prostoru v točko ${}^P\mathbf{p} \in \mathbb{R}^3$ na sliki je tako:

$${}^P\mathbf{p} \simeq \mathbf{S} \begin{bmatrix} {}^C\mathbf{R}_W & {}^C\mathbf{t}_W \end{bmatrix} {}^W\mathbf{p}. \quad (\text{B.12})$$

Upošteva se da je medsebojna lega dveh kamer v prostoru dana s povezavo ${}^{C_1}\mathbf{p} = {}^{C_1}\mathbf{R}_{C_2} {}^{C_2}\mathbf{p} + {}^{C_1}\mathbf{t}_{C_2}$ lahko izpeljemo medsebojno preslikavo med slikama kamer.

$${}^{C_1}\mathbf{p} = {}^{C_1}\mathbf{R}_{C_2} {}^{C_2}\mathbf{p} + {}^{C_1}\mathbf{t}_{C_2}, \quad (\text{B.13})$$

$$\mathbf{S}^{-1} {}^{P_1}\mathbf{p} \simeq {}^{C_1}\mathbf{R}_{C_2} \mathbf{S}^{-1} {}^{P_2}\mathbf{p} + {}^{C_1}\mathbf{t}_{C_2}, \quad (\text{B.14})$$

$$[{}^{C_1}\mathbf{t}_{C_2}]_{\times} \mathbf{S}^{-1} {}^{P_1}\mathbf{p} \simeq [{}^{C_1}\mathbf{t}_{C_2}]_{\times} {}^{C_1}\mathbf{R}_{C_2} \mathbf{S}^{-1} {}^{P_2}\mathbf{p}, \quad (\text{B.15})$$

$$0 = {}^{P_1}\mathbf{p}^T \mathbf{S}^{-T} [{}^{C_1}\mathbf{t}_{C_2}]_{\times} {}^{C_1}\mathbf{R}_{C_2} \mathbf{S}^{-1} {}^{P_2}\mathbf{p}, \quad (\text{B.16})$$

$$0 = {}^{P_1}\mathbf{p}^T \mathbf{F} {}^{P_2}\mathbf{p}. \quad (\text{B.17})$$

Preslikava med dvema slikama (B.17) je opisana s t. i. *fundamentalno matriko* (angl. „*fundamental matrix*“) $\mathbf{F}$.

Obravnavajmo sedaj primer, ko se vse točke v prostoru nahajajo v isti ravnini. Predpostavimo, da se vse točke nahajajo v ravnini ${}^Wz = 0$, kjer postavitev koordinatnega sistema W glede na koordinatni sistem C podaja transformacijska matrika ${}^C\mathbf{T}_W \in \text{SE}(3)$. Vsako točko v ravnini ${}^Wz = 0$ lahko opišemo z dvema koordinatama ${}^W\mathbf{p} \in \mathbb{R}^3$, zato lahko sistem (B.12) nekoliko poenostavimo:

$${}^P\mathbf{p} \simeq \mathbf{S} \begin{bmatrix} {}^C\mathbf{r}_{W,1} & {}^C\mathbf{r}_{W,2} & {}^C\mathbf{t}_W \end{bmatrix} {}^W\mathbf{p} = \mathbf{SZ} {}^W\mathbf{p} = \mathbf{H} {}^W\mathbf{p}, \quad (\text{B.18})$$

kjer je ${}^C\mathbf{r}_{W,i}$ i -ti stolpec rotacijske matrike ${}^C\mathbf{R}_W$. Matrika $\mathbf{Z}$ opisuje zunanje parametre kamere oz. medsebojno lego ravnin, ki skupaj z matriko notranjih parametrov $\mathbf{S}$ tvori matriko $\mathbf{H} \in \mathbb{R}^3 \times \mathbb{R}^3$, ki opisuje transformacijo med ravnino v prostoru in slikovno ravnino.

C Diskretna linearna transformacija (DLT) za ravnino

V nadaljevanju je predstavljena izpeljava *diskretne linearne transformacije* (DLT), ki je prirejena za ravninski primer. Izpeljava splošne DLT je opisana v [38].

Imamo dve množici ravninskih točk podanih v homogenih koordinatah: množico $\mathcal{A} = \{\underline{A}\mathbf{p}_i\}$, kjer je $\underline{A}\mathbf{p}_i = [x_i \ y_i \ 1]^T$, in množico $\mathcal{B} = \{\underline{B}\mathbf{p}_i\}$, kjer je $\underline{B}\mathbf{p}_i = [x_i \ y_i \ 1]^T$; $i = 1, \dots, N$. Vsaki točki iz množice $\mathcal{A}$ ustreza natanko ena in samo ena točka v množici $\mathcal{B}$, kar pomeni da sta moči obeh množic enaki, $N = |\mathcal{A}| = |\mathcal{B}|$. Vredno je poudariti, da so točke v obeh množicah podane v množicam lastnih lokalnih koordinatnih sistemih. Iščemo preslikavo $\mathbf{H} \in \mathbb{R}^3 \times \mathbb{R}^3$ — ta preslikava je znana kot *homografija* (angl. „*homography*”) —, ki podaja povezavo med ravninama v katerih ležita množici točk $\mathcal{A}$ in $\mathcal{B}$ in je podana s predpisom $\underline{A}\mathbf{p} \mapsto \underline{B}\mathbf{p}$ [39, str. 36, 45]:

$$\underline{A}\mathbf{p} \simeq \mathbf{H} \underline{B}\mathbf{p}. \quad (\text{C.1})$$

Upoštevamo:

$$\mathbf{H}^T = \begin{bmatrix} h_1 & h_2 & h_3 \end{bmatrix} \quad (\text{C.2})$$

in iz enačbe (C.1) izpeljemo naslednji relaciji:

$$A_x = \frac{h_1^T \underline{B}\mathbf{p}}{h_3^T \underline{B}\mathbf{p}} \quad \text{in} \quad A_y = \frac{h_2^T \underline{B}\mathbf{p}}{h_3^T \underline{B}\mathbf{p}}. \quad (\text{C.3})$$

Slednje lahko prepišemo v obliko:

$$\Psi \mathbf{h} = \begin{bmatrix} \underline{B}\mathbf{p}^T & \mathbf{0}^T & -A_x \underline{B}\mathbf{p}^T \\ \mathbf{0}^T & \underline{B}\mathbf{p}^T & -A_y \underline{B}\mathbf{p}^T \end{bmatrix} \mathbf{h} = \mathbf{0}, \quad (\text{C.4})$$

kjer je vektor $\mathbf{h}^T = [\mathbf{h}_1^T \mathbf{h}_2^T \mathbf{h}_3^T]$. Matrika $\mathbf{H}$ ima 8 prostostnih stopenj, zato je za rešljivost sistema (C.4) potrebno uvesti dodatno omejitev. Za enolično rešitev sistema (C.4) potrebujemo štiri nekolinearne točke ($N = 4$), vendar ker položaji parov točk nikoli niso natančno znani, za oceno matrike $\mathbf{H}$ običajno izberemo čim več parov točk ($N > 4$) in sistem (C.5) rešimo v smislu najmanjših kvadratov.

Predpostavimo lahko, da je vrednost elementa $h_{33} = 1$, kar nam omogoča, da — ob upoštevanju $\mathbf{h}^T = [\mathbf{h}_0^T \ 1]$ — zapišemo naslednji sistem enačb:

$$\Theta \mathbf{h}_0 = \begin{bmatrix} x_{b,1} & 0 & \dots & x_{b,N} & 0 \\ y_{b,1} & 0 & \dots & y_{b,N} & 0 \\ 1 & 0 & \dots & 1 & 0 \\ 0 & x_{b,1} & \dots & 0 & x_{b,N} \\ 0 & y_{b,1} & \dots & 0 & y_{b,N} \\ 0 & 1 & \dots & 0 & 1 \\ -x_{a,1}x_{b,1} & -y_{a,1}x_{b,1} & \dots & -x_{a,N}x_{b,N} & -y_{a,N}x_{b,N} \\ -x_{a,1}y_{b,1} & -y_{a,1}y_{b,1} & \dots & -x_{a,N}y_{b,N} & -y_{a,N}y_{b,N} \end{bmatrix}^T \begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{bmatrix} = \begin{bmatrix} x_{a,1} \\ y_{a,1} \\ \vdots \\ x_{a,N} \\ y_{a,N} \end{bmatrix} = \boldsymbol{\xi}, \quad (\text{C.5})$$

katerega rešitev lahko za $N \geq 4$ poiščemo z metodo najmanjših kvadratov:

$$\mathbf{h}_0 = (\Theta^T \Theta)^{-1} \Theta^T \boldsymbol{\xi}. \quad (\text{C.6})$$

Namesto omejitve ene vrednosti elementa v matriki $\mathbf{H}$ je običajno bolje vpeljati omejitev $\mathbf{h}^T \mathbf{h} = 1$, ki daje bolj robustno oceno. Rešitev je znana kot lastni vektor matrike $\Psi^T \Psi$, ki ustreza najmanjši lastni vrednoti te matrike. Rešitev lahko poiščemo s SVD razcepom, pri čemer je rešitev t. i. *desni singularni vektor* matrike Ψ .

Da izboljšamo numerično stabilnost ocene, se poslužujemo uporabe normiranja podatkov, ki ga napravimo pred izračunom ocene. Nad množico $\mathcal{A}$ uporabimo preslikavo $\mathcal{A} \rightarrow \mathcal{A}_n$ s predpisom $\underline{\mathbf{p}} \mapsto {}^A \underline{\mathbf{p}} = {}^A \mathbf{T}_A \underline{\mathbf{p}}$. Podobno preslikavo naredimo tudi na množici $\mathcal{B}$: $\mathcal{B} \rightarrow \mathcal{B}_n$ s predpisom $\underline{\mathbf{p}} \mapsto {}^{B_n} \underline{\mathbf{p}} = {}^{B_n} \mathbf{T}_B \underline{\mathbf{p}}$. Za način preslikave se pogosto uporablja *središčni razteg* (glej dodatek D). Najprej na podlagi točk v množicah $\mathcal{A}_n$ in $\mathcal{B}_n$ ocenimo t. i. normirane homografije $\mathbf{H}_n$, ki opisuje normirani sistem ${}^A \underline{\mathbf{p}} \simeq \mathbf{H}_n {}^{B_n} \underline{\mathbf{p}}$. Velja:

$${}^A \underline{\mathbf{p}} = {}^A \mathbf{T}_A^{-1} {}^A \underline{\mathbf{p}} \simeq {}^A \mathbf{T}_A^{-1} \mathbf{H}_n {}^{B_n} \mathbf{T}_B {}^{B_n} \underline{\mathbf{p}}. \quad (\text{C.7})$$

Zatorej lahko normirano homografijo $\boldsymbol{H}_n$ na koncu preslikamo v dejansko (iskano) homografijo s preslikavo:

$$\boldsymbol{H} \simeq {}^{A_n}\boldsymbol{T}_A^{-1} \boldsymbol{H}_n {}^{B_n}\boldsymbol{T}_B. \quad (\text{C.8})$$

D Središčni razteg

Preslikavo središčnega raztega predstavljata *premik* in *razteg*, ki zagotovita želeno srednjo vrednost množice in povprečno oddaljenost točk od središča množice. Transformacijska matrika za središčni razteg ima naslednjo obliko:

$$\mathbf{T} = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \\ 0 & 0 & 1 \end{bmatrix}, \quad (\text{D.1})$$

ki preslika množico točk $\mathcal{P} : \mathcal{P} \rightarrow \mathcal{P}_n$ z močjo $|\mathcal{P}| = N$ po predpisu $\underline{\mathbf{p}} \mapsto \underline{\mathbf{p}}_n = \mathbf{T}\underline{\mathbf{p}}$, $\underline{\mathbf{p}}^T = [x \ y \ 1]$.

Vrednosti parametrov raztega s in premika (t_x, t_y) izberemo npr. tako, da je srednja vrednost množice točk $\mathcal{P}_n$ enaka nič in povprečna oddaljenost od središča enaka $\sqrt{2}$ [71]. Iz zahteve:

$$\bar{\mathbf{p}}_n^T = [\bar{x}_n \ \bar{y}_n \ 1] = [s\bar{x} + t_x \ s\bar{y} + t_y \ 1] = [0 \ 0 \ 1] \quad (\text{D.2})$$

dobimo zvezi $t_x = -s\bar{x}$ in $t_y = -s\bar{y}$. Ker mora veljati:

$$\frac{1}{N} \sum_{i=1}^N \sqrt{x_{n,i}^2 + y_{n,i}^2} = \frac{s}{N} \sum_{i=1}^N \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2} = \sqrt{2}, \quad (\text{D.3})$$

dobimo še relacijo za faktor raztega:

$$s = \frac{N\sqrt{2}}{\sum_{i=1}^N \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2}}. \quad (\text{D.4})$$

Literatura

- [1] Corke, P. I. Visual control of robot manipulators — a review. *Visual Servoing*, zv. 7, str. 1–31, 1993.
- [2] Corke, P. I. *Visual control of robots: high-performance visual servoing*. Wiley, New York, 1996, 353 str.
- [3] Corke, P. *Robotics, vision and control: fundamental algorithms in MATLAB*. Springer, Berlin Heidelberg, 2011, 570 str.
- [4] Chaumette, F. in Hutchinson, S. Visual servo control, part I: basic approaches. *IEEE Robotics and Automation Magazine*, zv. 13, št. 4, str. 82–90, 2006.
- [5] Chaumette, F. in Hutchinson, S. Visual servo control, part II: advanced approaches. *IEEE Robotics and Automation Magazine*, zv. 14, št. 1, str. 109–118, 2007.
- [6] Oriolo, G., De Luca, A. in Vendittelli, M. WMR control via dynamic feedback linearization: design, implementation, and experimental validation. *IEEE Transactions on Control Systems Technology*, zv. 10, št. 6, str. 835–852, 2002.
- [7] Malis, E., Chesi, G. in Cipolla, R. 2 1/2 D visual servoing with respect to planar contours having complex and unknown shapes. *The International Journal of Robotics Research*, zv. 22, št. 10, str. 841–853, 2003.
- [8] Malis, E. Visual servoing invariant to changes in camera-intrinsic parameters. *IEEE Transactions on Robotics and Automation*, zv. 20, št. 1, str. 72–81, 2004.
- [9] Malis, E. in Vargas, M. Deeper understanding of the homography decompo-

- sition for vision-based control. Tehn. por., INRIA, Nice, France, 2007.
- [10] Kragic, D. in Christensen, H. I. Survey on visual servoing for manipulation. Tehn. por., Centre for Autonomous Systems, Numerical Analysis and Computer Science, Stockholm, Sweden, 2002.
- [11] Julier, S. J. *Process models for the navigation of high-speed land vehicles*. Doktorsko delo, University of Oxford, 1997.
- [12] Cherubini, A., Chaumette, F. in Oriolo, G. Visual servoing for path reaching with nonholonomic robots. *Robotica*, zv. 29, št. 07, str. 1037–1048, 2011.
- [13] Cherubini, A. in Chaumette, F. Visual navigation of a mobile robot with laser-based collision avoidance. *The International Journal of Robotics Research*, zv. 32, št. 2, str. 189–205, 2012.
- [14] Rao, R., Kumar, V. in Taylor, C. J. Visual servoing of a UGV from a UAV using differential flatness. V *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, zv. 1, str. 743–748. IEEE, 2003.
- [15] Rao, R. S., Kumar, V. in Taylor, C. J. Planning and control of mobile robots in image space from overhead cameras. V *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April, str. 2185–2190. IEEE, Barcelona, Spain, 2005.
- [16] Klančar, G. in Škrjanc, I. Tracking-error model-based predictive control for mobile robots in real time. *Robotics and Autonomous Systems*, zv. 55, št. 6, str. 460–469, 2007.
- [17] Diosi, A., Šegvić, S. in sod. Experimental evaluation of autonomous driving based on visual memory and image-based visual servoing. *IEEE Transactions on Intelligent Transportation Systems*, zv. 12, št. 3, str. 870–883, 2011.
- [18] DeSouza, G. N. in Kak, A. C. Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, zv. 24, št. 2, str. 237–267, 2002.
- [19] Krajník, T. *Large-scale mobile robot navigation and map building*. Doktorsko delo, Czech Technical University in Prague, 2011.
- [20] von Hundelshausen, F. *Computer vision for autonomous mobile robots*.

Doktorsko delo, Freien Universität Berlin, 2004.

- [21] Shakernia, O., Ma, Y. in sod. Vision guided landing of an unmanned air vehicle. V *38th Conference on Decision and Control (CDC)*, str. 4143–4148. IEEE, Phoenix, AZ, USA, 1999.
- [22] Bourquardez, O. in Chaumette, F. Visual servoing of an airplane for alignment with respect to a runway. V *IEEE International Conference on Robotics and Automation (ICRA)*, zv. 07, str. 1330–1335. IEEE, Roma, Italy, 2007.
- [23] Lumbar, S., Dolanc, G. in sod. Automatic guidance of an aircraft using predictive control in a visual servoing scheme. V *18th IFAC Symposium on Automatic Control in Aerospace (ACA)*, str. 6. Nara, Japan, 2010.
- [24] Bošnjak, M., Matko, D. in Blažič, S. Quadcopter hovering using position-estimation information from inertial sensors and a high-delay video system. *Journal of Intelligent & Robotic Systems*, zv. 67, št. 1, str. 43–60, 2011.
- [25] Bošnjak, M., Matko, D. in Blažič, S. Quadcopter control using an on-board video system with off-board processing. *Robotics and Autonomous Systems*, zv. 60, št. 4, str. 657–667, 2012.
- [26] Iwatani, Y., Watanabe, K. in Hashimoto, K. Unmanned helicopter control via visual servoing with occlusion handling. V G. Chesi in K. Hashimoto (Uredniki), *Visual Servoing via Advanced Numerical Methods*, Lecture Notes in Control and Information Sciences, str. 361–374. Springer, London, 2010.
- [27] Wenzel, K. E., Rosset, P. in Zell, A. Low-cost visual tracking of a landing place and hovering flight control with a microcontroller. *Journal of Intelligent and Robotic Systems*, zv. 57, št. 1-4, str. 297–311, 2010.
- [28] Klančar, G., Blažič, S. in sod. Image-based attitude control of a remote sensing satellite. *Journal of Intelligent and Robotic Systems*, zv. 66, št. 3, str. 343–357, 2012.
- [29] Kelly, R., Carelli, R. in sod. Stable visual servoing of camera-in-hand robotic systems. *IEEE/ASME Transactions on Mechatronics*, zv. 5, št. 1, str. 39–48, 2000.

- [30] Malis, E. in Chaumette, F. 2 1/2 D visual servoing with respect to unknown objects through a new estimation scheme of camera displacement. *International Journal of Computer Vision*, zv. 37, št. 1, str. 79–97, 2000.
- [31] Chan, A. *Constraint-aware visual servoing for teaching practical robot motion*. Magistrsko delo, The University Of British Columbia, 2009.
- [32] Bachta, W., Renaud, P. in sod. Visual servoing for beating heart surgery. V G. Chesi in K. Hashimoto (Uredniki), *Visual Servoing via Advanced Numerical Methods*, Lecture Notes in Control and Information Sciences, str. 91–114. Springer, London, 2010.
- [33] Mebarki, R., Krupa, A. in Chaumette, F. 2-D ultrasound probe complete guidance by visual servoing using image moments. *IEEE Transactions on Robotics*, zv. 26, št. 2, str. 296–306, 2010.
- [34] Baeten, J., Donné, K. in sod. Autonomous fruit picking machine: a robotic apple harvester. V C. Laugier in R. Siegwart (Uredniki), *Field and Service Robotics, Springer Tracts in Advanced Robotics*, zv. 42, str. 531–539. Springer, Berlin Heidelberg, 2008.
- [35] Faugeras, O. *Three-dimensional Computer Vision: A Geometric Viewpoint*. MIT Press, 1993, 663 str.
- [36] Faugeras, O. in Luong, Q.-T. *The Geometry of Multiple Images: The Laws that Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. MIT Press, 2004, 644 str.
- [37] Ma, Y., Soatto, S. in sod. *An Invitation to 3-D Vision: From Images to Geometric Models, Interdisciplinary Applied Mathematics*, zv. 26. Springer New York, New York, NY, 2004, 526 str.
- [38] Hartley, R. in Zisserman, A. *Multiple view geometry*. Cambridge University Press, New York, USA, 2. izd., 2004, 655 str.
- [39] Forsyth, D. in Ponce, J. *Computer vision: a modern approach*. Prentice Hall, illustrate izd., 2003, 693 str.
- [40] Cyganek, B. in Siebert, J. P. *An introduction to 3D computer vision techniques and algorithms*. John Wiley & Sons, Chichester, UK, 2009, 483

str.

- [41] Bradski, G. in Kaehler, A. *Learning OpenCV*. O'Reilly, first edit izd., 2008, 555 str.
- [42] Thrun, S., Burgard, W. in Fox, D. *Probabilistic robotics*. MIT Press, 4. izd., 2005, 647 str.
- [43] Siegwart, R., Nourbakhsh, I. R. in Scaramuzza, D. *Autonomous Mobile Robots*. MIT Press, Cambridge, Massachusetts, 2. izd., 2004, 453 str.
- [44] Siciliano, B., Sciavicco, L. in sod. *Robotics: Modelling, Planning and Control*. Springer-Verlag, London, 2010, 632 str.
- [45] Baeten, J., de Schutter, J. in Schutter, J. *Integrated Visual Servoing and Force Control: The Task Frame Approach*. Springer Science & Business Media, 2003, 179 str.
- [46] Chesi, G. in Hashimoto, K. *Visual Servoing via Advanced Numerical Methods*. Springer-Verlag, Berlin Heidelberg, 2010, 398 str.
- [47] Garcia, G. J., Corrales, J. a. in sod. Survey of visual and force/tactile control of robots for physical interaction in Spain. *Sensors (Basel, Switzerland)*, zv. 9, št. 12, str. 9689–733, 2009.
- [48] Kristan, M. in Perš, J. A Bayes-spectral-entropy-based measure of camera focus using a discrete cosine transform. *Pattern Recognition Letters*, zv. 27, št. 13, str. 1431–1439, 2006.
- [49] Sengupta, S. *Issues in 3D reconstruction from multiple views*. Msc, Indian Institute of Technology Madras, 2009.
- [50] Dani, A. P. in Dixon, W. E. Single camera structure and motion estimation. V G. Chesi in K. Hashimoto (Uredniki), *Visual Servoing via Advanced Numerical Methods, Lecture Notes in Control and Information Sciences*, zv. 401, str. 209–229. Springer, London, 2010.
- [51] Izadi, S., Kim, D. in sod. Kinectfusion: real-time 3D reconstruction and interaction using a moving depth camera. V *Proceedings of the 24th annual ACM symposium on User interface software and technology*, str. 559–568. 2011.

- [52] Mure-dubois, J. in Heinz, H. Fusion of time of flight camera point clouds. *Time*, str. 1–12, 2008.
- [53] Shademan, A. in Janabi-Sharifi, F. Sensitivity analysis of EKF and iterated EKF pose estimation for position-based visual servoing. *Proceedings of 2005 IEEE Conference on Control Applications, 2005. CCA 2005.*, 2005.
- [54] Becerra, H. M. in Sagues, C. Dynamic pose-estimation from the epipolar geometry for visual servoing of mobile robots. *2011 IEEE International Conference on Robotics and Automation*, str. 417–422, 2011.
- [55] Janabi-Sharifi, F. in Marey, M. A Kalman-Filter-Based Method for Pose Estimation in Visual Servoing. *IEEE Transactions on Robotics*, zv. 26, 2010.
- [56] Fontanelli, D., Salaris, P. in sod. Unicycle-like Robots with Eye-in-Hand Monocular Cameras : From PBVS towards IBVS. V G. Chesi in K. Hashimoto (Uredniki), *Visual Servoing via Advanced Numerical Methods*, Lecture Notes in Control and Information Sciences, str. 335–360. Springer, London, 2010.
- [57] De Luca, A., Oriolo, G. in Giordano, P. R. Image-based visual servoing schemes for nonholonomic mobile manipulators. *Robotica*, zv. 25, št. 02, str. 131–145, 2007.
- [58] Bourquardez, O., Mahony, R. in sod. Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle. *IEEE Transactions on Robotics*, zv. 25, št. 3, str. 743–749, 2009.
- [59] Mariottini, G. L. *Image-based Visual Servoing for Mobile Robots: the Multiple View Geometry Approach*. Phd, University of Siena, 2005.
- [60] Malis, E., Chaumette, F. in Boudet, S. 2-1/2-D visual servoing. *IEEE Transactions on Robotics and Automation*, zv. 15, št. 2, str. 238–250, 1999.
- [61] Gans, N. R. in Hutchinson, S. A. Stable visual servoing through hybrid switched-system control. *IEEE Transactions on Robotics*, zv. 23, št. 3, str. 530–540, 2007.
- [62] Lippiello, V., Siciliano, B. in Villani, L. Eye-in-hand/eye-to-hand multi-camera visual servoing. *Matrix*, str. 5354–5359, 2005.

-
- [63] Kornuta, T. in Zieliński, C. Robot Control System Design Exemplified by Multi-Camera Visual Servoing. *Journal of Intelligent & Robotic Systems*, 2013.
- [64] Dame, A. in Marchand, E. Mutual information-based visual servoing. *IEEE Transactions on Robotics*, zv. 27, št. 5, str. 958–969, 2011.
- [65] Perš, J. in Kovačič, S. Nonparametric, model-based radial lens distortion correction using tilted camera assumption. V *Proceedings of the Computer Vision Winter Workshop*, str. 286–295. 2002.
- [66] Tardif, J.-P., Sturm, P. in sod. Calibration of cameras with radially symmetric distortion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, zv. 31, št. 9, str. 1552–1566, 2009.
- [67] Tsai, R. Y. A versatile camera calibration technique for high-accuracy 3D machine vision metrology using off-the-shelf TV cameras and lenses. *IEEE Journal on Robotics and Automation*, zv. 3, št. 4, str. 323–344, 1987.
- [68] Brown, D. C. Close-range camera calibration. *Photogrammetric engineering*, zv. 37, št. 8, str. 855–866, 1971.
- [69] Bouguet, J.-Y. *Visual methods for three-dimensional modeling*. Doktorsko delo, California Institute of Technology, 1999.
- [70] Zhang, Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, zv. 22, št. 11, str. 1330–1334, 2000.
- [71] Hartley, R. I. In defence of the 8-point algorithm. V *IEEE International Conference on Computer Vision*, str. 1064–1070. IEEE, 1995.
- [72] Klančar, G., Kristan, M. in sod. Robust and efficient vision system for group of cooperating mobile robots with application to soccer robots. *ISA transactions*, zv. 43, št. 3, str. 329–342, 2004.
- [73] Comport, A. I. *Towards a computer imagination: Robust real-time 3D tracking of rigid and articulated objects for augmented reality and robotics*. Doktorsko delo, L'Université de Rennes I, 2005.
- [74] Krivic, J. *Segmentacija in 3D sledenje objektom na osnovi superkvadričnih*

- modelov*. Doktorsko delo, University of Ljubljana, 2006.
- [75] Rosten, E. *High performance rigid body tracking*. Doktorsko delo, University of Cambridge, 2006.
- [76] Zimmermann, K. *Fast learnable methods for object tracking*. Doktorsko delo, Czech Technical University, 2008.
- [77] Kristan, M. *Sledenje ljudi v video posnetkih s pomočjo verjetnostnih modelov*. Doktorsko delo, Univerza v Ljubljani, 2008.
- [78] Malis, E., Hamel, T. in sod. Estimation of homography dynamics on the special linear group. V G. Chesi in K. Hashimoto (Uredniki), *Visual Servoing via Advanced Numerical Methods*, Lecture Notes in Control and Information Sciences, str. 133–150. Springer, London, 2010.
- [79] Brown, M. in Lowe, D. G. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, zv. 74, št. 1, str. 59–73, 2006.
- [80] Furukawa, Y. in Ponce, J. Accurate, dense, and robust multi-view stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, zv. 32, št. 8, str. 362–376, 2010.
- [81] Tomasi, C. in Kanatani, K. Detection and Tracking of Point Features. Tehn. por. 7597, Carnegie Mellon University, Pittsburg, 1991.
- [82] Lowe, D. G. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, zv. 60, št. 2, str. 91–110, 2004.
- [83] Bay, H., Ess, A. in sod. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, zv. 110, št. 3, str. 346–359, 2008.
- [84] Mikolajczyk, K. in Schmid, C. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, zv. 27, št. 10, str. 1615–1630, 2005.
- [85] Matas, J., Chum, O. in sod. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, zv. 22, št. 10, str. 761–767, 2004.
- [86] Smith, S. M. in Brady, J. M. SUSAN — a new approach to low level image

- processing. *International Journal of Computer Vision*, zv. 23, št. 1, str. 45–78, 1997.
- [87] Calonder, M., Lepetit, V. in sod. BRIEF: binary robust independent elementary features. V K. Daniilidis, P. Maragos in N. Paragios (Uredniki), *Computer Vision — ECCV 2010*, Lecture Notes in Computer Science, str. 778–792. Springer, Heraklion, Crete, Greece, 2010.
- [88] Rublee, E., Rabaud, V. in sod. ORB: an efficient alternative to SIFT or SURF. V *International Conference on Computer Vision (ICCV)*, str. 2564–2571. IEEE, Barcelona, Spain, 2011.
- [89] Alahi, A., Ortiz, R. in Vanderghenst, P. FREAK: fast retina keypoint. V *Conference on Computer Vision and Pattern Recognition (CVPR)*, str. 510–517. IEEE, Providence, RI, USA, 2012.
- [90] Torr, P. H. S. P. in Zisserman, A. MLESAC: a new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, zv. 78, št. 1, str. 138–156, 2000.
- [91] Vedaldi, A., Jin, H. in sod. KALMANSAC: robust filtering by consensus. V *Tenth IEEE International Conference on Computer Vision (ICCV)*, zv. 1, str. 633–640. IEEE, 2005.
- [92] Chum, O. in Matas, J. Matching with PROSAC — progressive sample consensus. V *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, I, str. 220–226. IEEE, 2005.
- [93] Choi, S., Kim, T. in Yu, W. Performance evaluation of RANSAC family. *Journal of Computer Vision*, zv. 24, št. 3, str. 271–300, 1997.
- [94] Fischler, M. A. in Bolles, R. C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, zv. 24, št. 6, str. 381–395, 1981.
- [95] Vargas, M. in Malis, E. Visual servoing based on an analytical homography decomposition. V *Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference*, 3, str. 5379–5384. IEEE, 2005.

- [96] López-Nicolás, G., Bhattacharya, S. in sod. Switched homography-based visual control of differential drive vehicles with field-of-view constraints. V *IEEE International Conference on Robotics and Automation*, April, str. 4238–4244. IEEE, Roma, Italy, 2007.
- [97] Sebastián, J. M., Pari, L. in sod. Uncalibrated visual servoing using the fundamental matrix. *Robotics and Autonomous Systems*, zv. 57, št. 1, str. 1–10, 2009.
- [98] Collewet, C. in Marchand, E. Photometric visual servoing. *IEEE Transactions on Robotics*, zv. 27, št. 4, str. 828–834, 2011.
- [99] Achkar, A. J. *Four ways to stabilize a nonholonomic system on the example of a mobile robot*. Msc, McGill University, Montreal, 1994.
- [100] Aicardi, M., Casalino, G. in sod. Closed loop steering of unicycle-like vehicles via Lyapunov techniques, 1995.
- [101] Ma, Y., Košecká, J. in Sastry, S. S. Vision guided navigation for a non-holonomic mobile robot. *IEEE Transactions on Robotics and Automation*, zv. 15, št. 3, str. 521–536, 1999.
- [102] Chwa, D. Sliding-mode tracking control of nonholonomic wheeled mobile robots in polar coordinates. *IEEE Transactions on Control Systems Technology*, zv. 12, št. 4, str. 637–644, 2004.
- [103] Chwa, D., Hong, S.-K. in Song, B. Robust posture stabilization of wheeled mobile robots in polar coordinates. V *Proceedings of the 17th International Symposium on Mathematical Theory of Networks and Systems*, str. 343–348. Kyoto, Japan, 2006.
- [104] Galicki, M. Nonholonomic motion planning of mobile. V K. R. Kozłowski (Urednik), *Robot Motion and Control 2009*, Lecture Notes in Control and Information Sciences, str. 277–286. Springer, London, 2009.
- [105] Blažič, S. A novel trajectory-tracking control law for wheeled mobile robots. *Robotics and Autonomous Systems*, zv. 59, št. 11, str. 1001–1007, 2011.
- [106] Brockett, R. W. Asymptotic stability and feedback stabilization. V R. S. Millman in H. J. Sussmann (Uredniki), *Differential Geometric Control*

-
- Theory*, str. 181–191. Birkhuser, Boston, MA, 1983.
- [107] LaValle, S. M. *Planning algorithms*. Cambridge University Press, 2006, 1007 str.
 - [108] Škrjanc, I. in Klančar, G. Optimal cooperative collision avoidance between multiple robots based on Bernstein-Bézier curves. *Robotics and Autonomous Systems*, zv. 58, št. 1, str. 1–9, 2010.
 - [109] Ning, K., Kulvicius, T. in sod. A novel trajectory generation method for robot control. *Journal of Intelligent and Robotic Systems*, zv. 68, št. 2, str. 165–184, 2012.
 - [110] Milam, M. B. *Real-time optimal trajectory generation for constrained dynamical systems*. Doktorsko delo, California Institute of Technology, 2003.
 - [111] Pozna, C., Troester, F. in sod. On the design of an obstacle avoiding trajectory: method and simulation. *Mathematics and Computers in Simulation*, zv. 79, št. 7, str. 2211–2226, 2009.
 - [112] Lapierre, L. in Zapata, R. A guaranteed obstacle avoidance guidance system. *Autonomous Robots*, zv. 32, št. 3, str. 177–187, 2012.
 - [113] Morari, M. in Lee, J. H. Model predictive control: past, present and future. *Computers and Chemical Engineering*, zv. 23, št. 4-5, str. 667–682, 1999.
 - [114] Maciejowski, J. M. *Predictive control: with constraints*. Pearson Education, 2002, 331 str.
 - [115] Brosilow, C. in Joseph, B. Two-degree of freedom internal model control. V *Techniques of Model-Based Control*, str. 65–84. Prentice Hall, 2002.
 - [116] Camacho, E. F. in Bordons, C. *Model predictive control*. Springer, 2. izd., 2004, 405 str.
 - [117] Rouhani, R. in Mehra, R. K. Model algorithmic control (MAC); basic theoretical properties. *Automatica*, zv. 18, št. 4, str. 401–414, 1982.
 - [118] Richalet, J. Industrial applications of model based predictive control. *Automatica*, zv. 29, št. 5, str. 1251–1274, 1993.
 - [119] Normey-Rico, J. E., Gómez-Ortega, J. in Camacho, E. F. A Smith-predictor-

- based generalised predictive controller for mobile robot path-tracking. *Control Engineering Practice*, zv. 7, št. 6, str. 729–740, 1999.
- [120] Škrjanc, I. in Matko, D. Predictive functional control based on fuzzy model for heat-exchanger pilot plant. *IEEE Transactions on Fuzzy Systems*, zv. 8, št. 6, str. 705–712, 2000.
- [121] Lepetič, M., Škrjanc, I. in sod. Predictive functional control based on fuzzy model: comparison with linear predictive functional control and PID control. *Journal of Intelligent and Robotic Systems*, zv. 36, št. 4, str. 467–480, 2003.
- [122] Richalet, J. in O'Donovan, D. *Predictive functional control: principles and industrial applications*. Springer, 2009, 222 str.
- [123] Teslić, L., Škrjanc, I. in Klančar, G. Using a LRF sensor in the Kalman-filtering-based localization of a mobile robot. *ISA transactions*, zv. 49, št. 1, str. 145–153, 2010.
- [124] Zhang, Z., Nejat, G. in sod. A novel 3D sensory system for robot-assisted mapping of cluttered urban search and rescue environments. *Intelligent Service Robotics*, zv. 4, št. 2, str. 119–134, 2011.
- [125] Civera, J., Davison, A. J. in Martínez Montiel, J. M. Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, zv. 24, št. 5, str. 932–945, 2008.
- [126] Andreasson, H. *Local visual feature based localisation and mapping by mobile robots*. Doktorsko delo, Örebro University, 2008.
- [127] Lee, Y.-J. in Song, J.-B. Three-dimensional iterative closest point-based outdoor SLAM using terrain classification. *Intelligent Service Robotics*, zv. 4, št. 2, str. 147–158, 2011.
- [128] Choset, H. *Principles of robot motion: theory, algorithms, and implementations*. MIT Press, illustrate izd., 2005, 603 str.
- [129] Gadeyne, K. *Sequential Monte Carlo methods for rigorous Bayesian modeling of autonomous compliant motion*. Doktorsko delo, Katholieke Universiteit Leuven, 2005.
- [130] Rosales, E. M. in Gan, Q. Forward and inverse kinematics models for a

-
- 5-DOF Pioneer 2 robot arm. Tehn. por. October, Department of Computer Science, University of Essex, 2002.
- [131] Stanciu, I. R. *Designing visually servoed tracking to augment camera teleoperators*. Doktorsko delo, Drexel University, 2004.
 - [132] Pomares Baeza, J. in García Gómez, Gabriel Jesús Torres Medina, F. A robust approach to control robot manipulators by fusing visual and force information. *Journal of Intelligent and Robotic Systems*, zv. 48, št. 4, str. 437–456, 2007.
 - [133] Hughes, C., Denny, P. in sod. Equidistant fish-eye calibration and rectification by vanishing point extraction. *IEEE transactions on pattern analysis and machine intelligence*, zv. 32, št. 12, str. 2289–96, 2010.
 - [134] Heikkila, J. Geometric camera calibration using circular control points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, zv. 22, št. 10, str. 1066–1077, 2000.
 - [135] Sugimoto, A. A linear algorithm for computing the homography from conics in correspondence. *Journal of Mathematical Imaging and Vision*, zv. 13, št. 2, str. 115–130, 2000.
 - [136] Mudigonda, P. K., Jawahar, C. V. in Narayanan, P. J. Geometric structure computation from conics. V *Proc. Indian Conf. Computer Vison, Graphics and Image Processing (ICVGIP)*, str. 9–14. 2004.
 - [137] Ip, H. H. in Chen, Y. Planar rectification by solving the intersection of two circles under 2D homography. *Pattern Recognition*, zv. 38, št. 7, str. 1117–1120, 2005.
 - [138] Wu, Y., Li, X. in sod. Coplanar circles, quasi-affine invariance and calibration. *Image and Vision Computing*, zv. 24, št. 4, str. 319–326, 2006.
 - [139] Lourakis, M. I. Plane metric rectification from a single view of multiple coplanar circles. V *2009 16th IEEE International Conference on Image Processing (ICIP)*, str. 509–512. IEEE, 2009.
 - [140] Zhang, Z. Parameter estimation techniques: a tutorial with application to conic fitting. *Image and vision Computing*, zv. 15, št. 1, str. 59–76, 1997.

-
- [141] Fitzgibbon, A., Pilu, M. in Fisher, R. B. Direct least square fitting of ellipses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, zv. 21, št. 5, str. 476–480, 1999.
- [142] Chernov, N. Ellipse fit (direct method), 2009.
- [143] Zdešar, A., Škrjanc, I. in Klančar, G. Homography estimation from circular motion for use in visual control. *Robotics and Autonomous Systems*, zv. 62, št. 10, str. 1486–1496, 2014.
- [144] Boykov, Y., Veksler, O. in Zabih, R. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, zv. 23, št. 11, str. 1222–1239, 2001.
- [145] Boykov, Y. in Kolmogorov, V. An Experimental Comparison of Min-Cut / Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, zv. 26, št. 9, str. 1124–1137, 2004.
- [146] Boykov, Y. in Veksler, O. Graph cuts in vision and graphics: Theories and applications. V N. Paragios, Y. Chen in O. Faugeras (Uredniki), *Handbook of mathematical models in computer vision*, str. 79–96. Springer, 2006.
- [147] Kolmogorov, V. in Zabih, R. Graph Cut Algorithms for Binocular Stereo with Occlusions. V N. Paragios, Y. Chen in O. Faugeras (Uredniki), *Handbook of Mathematical Models in Computer Vision*, pogl. 26, str. 423–437. Springer, 2006.
- [148] Felzenszwalb, P. F. in Zabih, R. Dynamic programming and graph algorithms in computer vision. *IEEE transactions on pattern analysis and machine intelligence*, zv. 33, št. 4, str. 721–40, 2011.
- [149] Danuser, G. Computer vision in cell biology. *Cell*, zv. 147, št. 5, str. 973–8, 2011.
- [150] Lowe, D. Object recognition from local scale-invariant features. *Proceedings of the Seventh IEEE International Conference on Computer Vision*, str. 1150–1157 vol.2, 1999.
- [151] The OpenCV reference manual: release 2.4.3, 2012.

-
- [152] Rosten, E. in Drummond, T. Fusing points and lines for high performance tracking. *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, str. 1508–1515 Vol. 2, 2005.
- [153] Mair, E., Hager, G. D. in sod. Adaptive and Generic Corner Detection Based on the Accelerated Segment Test. V *Computer Vision — ECCV 2010*, str. 183–196. Springer Berlin Heidelberg, 2010.
- [154] Dalal, N. in Triggs, B. Histograms of Oriented Gradients for Human Detection. V *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, zv. 1, str. 886–893. IEEE, San Diego, CA, USA, 2005.
- [155] Ke, Y. in Sukthankar, R. PCA-SIFT: a more distinctive representation for local image descriptors. V *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, str. 506–513. IEEE, 2004.
- [156] Leutenegger, S., Chli, M. in Siegwart, R. Y. BRISK: Binary Robust invariant scalable keypoints. *2011 International Conference on Computer Vision*, str. 2548–2555, 2011.
- [157] Barron, J. L. in Thacker, N. A. Tutorial: computing 2D and 3D optical flow. Tehn. por. 2004, University of Manchester, Manchester, 2005.
- [158] Herisse, B., Russotto, F.-x. in sod. Hovering flight and vertical landing control of a VTOL unmanned aerial vehicle using optical flow. *Image (Rochester, N.Y.)*, str. 22–26, 2008.
- [159] Scaramuzza, B. D. in Fraundorfer, F. Visual Odometry Part I: The First 30 Years and Fundamentals. *IEEE Robotics & Automation Magazine*, str. 80–92, 2011.
- [160] Hančič, M., Šink, A. in sod. Vodenje in lokalizacija mobilnega sistema v okolju. V *ERK*, slika 3, str. 1–4. IEEE, Portorož, Slovenija, 2014.
- [161] Klančar, G. *Avtonomni mobilni sistemi*. Fakulteta za elektrotehniko, Univerza v Ljubljani, Ljubljana, 2013, 223 str.
- [162] Nistér, D., Naroditsky, O. in Bergen, J. Visual odometry for ground vehicle

- applications. *Journal of Field Robotics*, zv. 23, št. 1, str. 3–20, 2006.
- [163] Scaramuzza, D., Fraundorfer, F. in Siegwart, R. Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC. *2009 IEEE International Conference on Robotics and Automation*, str. 4293–4299, 2009.
- [164] Civera, J., Grasa, O. G. in sod. 1-point RANSAC for extended Kalman filtering: application to real-time structure from motion and visual odometry. *Journal of Field Robotics*, zv. 27, št. 5, str. 609–631, 2010.
- [165] Fraundorfer, B. F. in Scaramuzza, D. Visual Odometry Part II: Matching, Robustness, Optimization, and Application. *IEEE Robotics & Automation Magazine*, str. 78–90, 2012.
- [166] Strasdat, H., Montiel, J. M. M. in Davison, A. J. Visual SLAM: Why filter? *Image and Vision Computing*, zv. 30, št. 2, str. 65–77, 2012.
- [167] Nistér, D., Naroditsky, O. in Bergen, J. Visual odometry. V *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'04)*, str. 652–659. 2004.
- [168] Blažič, S. On Periodic Control Laws for Mobile Robots. *IEEE Transactions on Industrial Electronics*, zv. 61, št. 7, str. 3660–3670, 2014.
- [169] Precup, R.-E., Preitl, S. in sod. Generic two-degree-of-freedom linear and fuzzy controllers for integral processes. *Journal of the Franklin Institute*, zv. 346, št. 10, str. 980–1003, 2009.
- [170] Piegl, L. in Tiller, W. *The NURBS book*. Springer-Verlag Berlin Heidelberg, 2. izd., 1997.
- [171] Farin, G. From Conies to NURBS: A tutorial and survey. *IEEE Computer Graphics and applications*, zv. 12, št. 5, str. 78–86, 1992.
- [172] Phillips, G. M. A de Castelja algorithm for generalized bernstein polynomials. *BIT Numerical Mathematics*, zv. 37, št. 1, str. 232–236, 1997.
- [173] Hanson, A. J. *Visualizing quaternions*. Elsevier, San Francisco, CA, USA, 2006, 498 str.