

Doctoral Dissertation

博士論文

Image-Based and Position-Based Visual Servo for Automated Vehicles and Robots: Integrated Design of Control, Sensing, and Estimation

(自律車両とロボットの為の画像ベースと位置ベースのビジュアル
サーボ：制御・センシング・推定器の統合的な設計)

by

37-177075 **Yoshi Ri**
李 堯希

Dissertation Submitted to

Department of Electrical Engineering

for the Degree of

Doctor of Philosophy

at

The University of Tokyo

29 November 2019

Supervisor:

Professor Hiroshi Fujimoto

Abstract

Since image sensors such as cameras using visible light are lightweight, inexpensive, and can provide abundant information, they play a vital role in autonomous driving robots and autonomous mobile vehicles requiring environmental recognition. There are well-known matters in visual sensing for the robotics: ambiguity on depth and recognition. camera projection And image-based autonomous control contains many elements such as sensors, image processing on optical projection, and filter design to estimate necessary information such as depth. Since these elements are intricately intertwined in control systems, integrated design considering all of them are required. In this paper, was carried out position control of a robot arm and a small vehicle on a base of a method to guide a robot on a base of image information called visual servo.

This paper discusses two methods according to the classifications of a visual servo, which are called image-based and position-based method. The image-based visual servo is a technique to control the position with the two-dimensional image feature, such as the feature point from the image acquired from the camera. And the position-based method is a technique to estimate the three-dimensional position from the image and to control based on it. The former can be controlled without requiring distance information, and is robust against modeling errors such as calibration, but has a disadvantage that it cannot control three-dimensional behavior. In the latter, though the control law is clear and easy to handle, the problem is that difficult distance estimation must be carried out. Chapter 2–4 summarizes the image-based approach mainly using industrial robot arms, and Chapter 5–8 summarizes the position-based approach toward autonomous mobile robots.

In chapter 2, we proposed a 2D FFT based image displacement estimation method, which is highly accurate and false-aware, and its tuning guidelines. The performance of the proposed method was evaluated by the comparison with the conventional feature-point-based method. The proposed method is superior in estimation on vague images, constant computation time, and false-awareness.

In chapter 3, we designed an image-based visual servo system based on the features obtained from the sensing method proposed in chapter 2 and demonstrate its effectiveness in experiments. Then, the proposed coordinate transformation leads to the image jacobian to be time-invariant matrix when targeting a planar object. Therefore the feedforward control can be easily realized with proper scaling estimation, and it can greatly improve tracking performance. This visual tracking can be applied to the teaching of the operation of a robot based on the video information.

In chapter 4, we examined a method to accurately extract the camera motion from the video used in the visual tracking in Chapter 3. The proposed coordinate transformation, which is similar in chapter 3, can transform the image displacements optimization problem to simple linear least-squares. And we proposed a computationally efficient method by introducing the concept of a distance matrix. It is shown that the control law and estimator can be designed based on a suitable sensing method based on the fact

that the correct weight can be obtained by the false-aware estimation proposed in chapter 2.

In chapter 5, depth estimation utilizing the height constraint of the ground vehicle and coarse/fine marker tracking that combines template tracking and color information was introduced for indoor position control of the ground vehicle. As a result, a millimeter-order-high-precision position estimation with small markers was achieved.

Chapter 6 described a distance estimation method using a stereo camera. To solve the stereo accuracy and range tradeoff, we proposed the sensor fusion technique to estimate robust and accurate depth with stereo disparity and relative scaling from monocular vision. The improvement of the responsiveness of estimation, which has been a bottleneck, is reported by proposing a switching observer using a pole arrangement instead of the commonly used extended kalman filter.

Chapter 7 discusses the evaluation of sensing and state estimator design in chapter 6 when applied to adaptive cruise control. Using actual parameters in ACC, we showed that sensor fusion based on pole placement is useful in high-gain cases where the responsiveness of distance control between vehicles is enhanced. This insists that we have to evaluate the observation not with its mean error or variance but with final control performance, including convergence speed.

In chapter 8, a theoretical calculation method using the Lyapunov equation was used to determine how sensor noise from sensors and image processing affects state estimates, state variables, and final output. This made it possible to design an estimator that reduces errors in a steady state in a control system when sensor noise and sampling are determined. In addition, by fixing the controller and the estimator, we could also select an appropriate image processing algorithm from the system information in advance.

In the image-based methods in Chapters 2 to 4, it was shown that the control method is affected by determining the sensing method, and necessary estimators such as the scale amount are also affected by this. Therefore, it became clear that it is important to select the algorithm used for sensing so that it is convenient for control and estimation. The improvement of trajectory tracking performance can be expected by feedforward, which is usually difficult with a series of visual servos based on the phase correlation proposed in this paper.

Each sensing, estimation, and control method in this paper not only solves the particular problems but also applicable to other robotics application. And the methodology to choose proper combination is already explained. The robotics in the near future will require such cross-disciplinary thinking; we hope that this study will make a contribution to the development of robotics technology.

Abstract(概要)

可視光を用いたカメラなどの画像センサは軽量・安価でかつ豊富な情報量を提供できるため、環境認識が必要な自律駆動ロボットや自律移動車両において非常に重要な役割を担っている。これらにおいて、画像という2次元平面上に3次元空間の情報を投影する性質上、カメラの距離方向に関する曖昧さが残る点、画像という2次元の輝度の配列から意味のある情報を抽出しなければならない点などの問題がよく知られている。また、画像情報を用いた自律制御には光学的な投影に関するモデリングや画像処理などのセンシング、そこから更に必要な情報を抽出する推定、そして得られた情報を元にロボットや車両を制御する要素が複雑に絡み合うため、それら全てを考慮した統合的な開発・設計が必要となる。

本論文ではビジュアルサーボと呼ばれる画像情報を元にロボットを誘導する手法を元にロボットアームや小型車両の位置制御を行う。

本論文はビジュアルサーボの分類に従い、大まかに画像ベースと位置ベースの2つの手法に分けて議論を行う。画像ベースとはカメラから取得した画像から特徴点等の2次元の特徴量を元に位置制御を行う手法であり、他方で位置ベースとは画像から3次元的な位置などを推定してそれを基に制御を行う手法である。前者は距離情報を必要とせずに制御でき、キャリブレーションなどのモデル化誤差にも頑強だが、3次元的な挙動を制御しきれない欠点がある。後者は制御則が明快で扱いやすいが、困難な距離推定を行わなければならない点の問題である。2～4章では主に産業用ロボットアームを用いた画像ベースのアプローチ、5～8章では自律移動ロボットに向けた位置ベースのアプローチについてまとめる。

まず、2章では画像から平行移動や回転、拡大縮小などの幾何的な変位を抽出する際に問題となる不確実性の評価や精度向上を目指すために2次元フーリエ変換に基づく位相相関と呼ばれるロバストな変位計測手法の改善について提案を行う。画像を2次元フーリエ変換した周波数領域において位相と振幅のそれぞれに平行移動成分と回転・拡大縮小成分の情報が含まれており、適切なフィルタリングを施すことに依って変位を抽出できる。精度やロバスト性に影響するフィルタや各種変数の決定手法を明示した上で一般的によく用いられるSIFTやSURFなどの特徴点を用いた手法との精度・ロバスト性比較を行った結果、特定のパターンや計算時間、検出の正誤判定などの有用性において利点があることを示した。

次に第3章では2章で検出した2次元の変位をフィードバックして軌道追従制御を行う手法を提案した。画像ベースのビジュアルサーボにおいては画像内の動きと3次元での動きを対応付けるイメージャコビアン の計算が必要であったが、センシングに合わせた制御則を用いることでこのイメージャコビアンを時不変行列に変換できることを示した。その結果、本来は時々刻々と変化する距離情報を元に計算する必要があるため現実的でなかったフィードフォワード制御を導入することができ、高次の軌道に対する追従性を大きく向上させることができる。

続く第4章では3章で行った軌道追従の指令値生成の高精度化を行った。画像ベースのビジュアルサーボにおける指令値とは連続的な画像すなわち動画であり、3章の軌道追従制御では動画で定義された教示動作を基にロボットの動作再現が可能である。その際に動画の連続するフレーム間の変位情報を元に指令値を生成するが、指令値に用いる複数のフレーム間の変位情報を元に最適化を行うことでより精密な動作再現を可能にする。距離行列を基にした効率の良い最適化アルゴリズムを提案したことでより大規模な軌道追従制御にて有効な手

法を提案した。

第5章からは位置ベースの手法にて問題になる距離方向の曖昧さを解決する手段として物体の位置制約を用いた位置推定手法を提案した。小型の自律移動車両の実験のための室内の実験フィールドにおけるGPSの代わりの位置センサとして天井付近に設置したカメラを用いており、車両ロボットの背面に取り付けたマーカの高さが変化しないという制約を用いることでステレオ機構を用いた3次元ビジョンセンサよりも高精度に位置を推定できることを示し、実験用の環境を整えた。

続く第6章ではステレオカメラを用いた距離推定手法を取り扱う。ステレオカメラでは隣接するカメラより得られる視差情報を元に距離を推定しているが、物体が至近距離にある場合などでは視差が十分に取れない問題がある。したがって、単眼でも得られる物体のスケール変化に関する観測とステレオの視差と組み合わせることでロバストに距離推定を行う手法について提案を行う。また、その際の拡張カルマンフィルタや線形化オブザーバにおいて切り替え機構を適用することでより早く正確な推定が可能になることを確認した。

第7章では6章で得られる距離や距離方向の速度などの情報を元に Adaptive Cruise Control (ACC: 自動追従制御) を実現するための推定器の設計を行う。ACCにおける車両と車間距離の制御ではストリングスタビリティと呼ばれる車群の挙動の安定性が必要になるため、制御器にはある制約が課される。しかるに安定的に車間を制御するための距離推定器を設計する必要がある。現在のACCでは車車間通信による全車の速度加速度情報を前提としていることが多いため画像情報を元に自律的に安定的な制御ができれば大いに課題解決に役立つ。

第8章ではガウス分布を仮定したセンサの誤差と推定の収束速度の両方の側面から推定器を設計する手法について考察と提案を行う。制御で用いられるオブザーバは周波数特性や収束性を元に設計されるが、線形モデルとガウス分布の誤差を仮定することでリアプノフ方程式からセンサ誤差の伝搬度合いを事前に計算することが可能である。この枠組を用いることにより、推定器の設計結果からセンシングに用いる画像処理アルゴリズムの選定や画像処理アルゴリズムから所望の収束速度を満たす推定器の設計を行えることを示す。また、収束速度制約下での誤差分散最小化推定の問題は拡張すると行列不等式の形で記述されたH2ノルムの最適化問題と同系の双線形最適化問題に帰着できることを示した。

2～4章の画像ベースの手法ではセンシングの手段を決定することで制御手法が左右され、スケール量などの必要な推定器もこれに影響されることを示した。したがって、制御や推定において都合の良いようにセンシングに用いるアルゴリズムを選択することの重要性が明らかになった。本論文で提案した位相相関を基にした一連のビジュアルサーボでは通常困難なフィードフォワードによる軌道追従性能の向上が見込める。

5～8章の位置ベースの手法でも同様に主に誤差や推定の収束速度の観点からセンシング手段と推定器の協調が必要であることを示し、一連のセンシングアルゴリズム、オブザーバ等推定器、制御器の統合的な設計が必要であることを示した。一般的にはセンサや画像処理アルゴリズム有りきで制御などの設計を行っていたが、この枠組が浸透することに依って制御技術者の目指す制御目標を元に逆にセンサや処理アルゴリズムの設計を養成できる点で革新が期待される。

本論文における各々のセンシング・推定・制御手法が他のロボット・制御技術者の問題を解決するだけでなく、統合的設計の概念の浸透と分野間の連携強化によりロボット・制御技術にさらなる躍進がもたらされることを願う。

Acknowledgment(謝辞)

本研究をすすめるにあたり、毎週の報告会等において熱心なご指導と的確なご助言を頂いた藤本博志 准教授に心から感謝しております。学部生から博士後期課程までの6年間大変お世話になりました。制御工学周りの知識だけでなく、研究者として新しい物事に向かっていく姿勢には大きく影響を受けた自負があります。学問に真摯かつ積極的なその姿勢は今後も自分の中で目標にし続けていきたいと思っています。また、藤本先生のご紹介を通して触れた多くの研究プロジェクトや技術交流に触れられた研究生活は大変でしたがとても実りの多いものだったと実感しています。今後、本研究室と藤本先生の元で培った経験を社会に還元していきたいと思っています。これからどうぞよろしくお願いします。

研究室発表会等のご場でご指導ご鞭撻頂いた堀洋一教授に心から御礼申し上げます。研究相談こそあまりしませんでした。飲み会や面談でお話するたびに堀先生の築き上げられた人生観に自分もかなり影響されました。生活を律するのにはまだまだ改善が必要そうですが、最適化できないなりに人生そして研究を一生懸命やっています。

また、お忙しい中、私の博士審査に携わっていただいた古関先生、坂井先生、橋本先生に御礼申し上げます。先生方の建設的かつ鋭い貴重なご意見により、博士論文の質を向上することができました。

藤本先生や研究室を通して得られた貴重な経験は多くありますが、特に博士1年夏のNikon Research Corporation of America (NRCA) へのインターンと2年時から3年時末までのCRESTの活動について謝辞を述べさせていただきます。NRCAへのインターンを斡旋・ご助言頂いた当時先輩だった大西助教授には日頃から研究面や研究室生活で大いにお世話になり、感謝してもしきれません。およそ80日間という短いインターンでしたが非常に密度の濃い経験ができたと思っています。Nikonの馬込様やBausan様、その他現地のスタッフ様には大変お世話になりました。どうもありがとうございました。CRESTの無線電力伝送の社会実装に関する活動ではメンバーの先生方や清水助教授、柏ITS委員会の関係者の方々など多くの方にお世話になりました。ありがとうございました。

博士課程の3年間は東大工学系：SEUT-RA（博士課程学生特別リサーチ・アシスタント）の経済的援助により成立しました。電気系事務室の山田様には毎週とてもお世話になりました。

藤本研の秘書の松嶋様、植野様、今泉様には事務手続きや予算の相談まで大変お世話になりました。ありがとうございました。

また、研究室で研究をするにあたって、たくさんの研究室の先輩や同期・後輩の皆様にお世話になりました。日々の研究相談や勉強、余暇や研究室での雑談など含めてどれも楽しい時間でした。個別に謝辞を述べるときがありませんが、特にNanoチームの先輩としてお世話になった大西様、山田様、矢崎様に感謝申し上げます。また、6年間楽しさと辛さを分かち合った同期の延命君にも大変感謝申し上げます。

最後に、生まれてから9年の大学生活に至るまで、愛情を持って支え続けてくれた両親に心から感謝いたします。

Contents

Chapter 1	Introduction	1
1.1	Background	1
1.2	Summery of each chapter	2
Part I	Image-Based Method	5
Chapter 2	Practical Algorithm Design for the FFT-based Robust Image Registration Method	6
2.1	Motivation	6
2.2	The principle of displacement estimation via Phase Correlation	7
2.2.1	Translation Estimation	7
2.2.2	Rotation and Scaling Estimation	9
2.3	Proposed Algorithm	9
2.3.1	Window Function	10
2.3.2	Whitening	10
2.3.3	Scaling Parameter Desition for Log-polar Transformation	10
2.3.4	Bandpass Filtering for Rotation-Scaling Estimation	10
2.3.5	Refinement of Rotation and Scaling Estimation	12
2.3.6	Subpixel Matching	13
2.3.7	Estimation Varidation	13
2.4	Accuracy and Robustness Evaluation	18
2.4.1	Data set for Evaluation	18
2.4.2	Performance Comparsion	21
2.4.3	Comparison with Feature Points-based Method	21
2.5	Conclusion	23
Chapter 3	POC based Visual Servo and Reference Tracking	24
3.1	Motivation	24
3.2	Image Based Visual Servo	24
3.3	Reference Extraction from Video Frames	25
3.3.1	Reference Trajectory Description Based on Key Frame	26
3.3.2	Reference Concussion Using Relay Images	26
3.4	Video Tracking Control Using Image Based Visual Servo Scheme	26
3.4.1	Video Tracking with Feedforward Method	26
3.4.2	Derivation of Time-Invariant Image Jacobian [1]	27

	3.4.3	Comparison in Simulation	28
3.5		Distance Estimation for Image Jacobian Estimation	29
	3.5.1	Method 1: Scaling Parameter Based	29
	3.5.2	Method 2: Translation Parameter Based	29
	3.5.3	Distance Estimation Experiment	30
3.6		Experiment	30
	3.6.1	Reference Shaping and Filtering	32
	3.6.2	Experiment and Evaluation	32
3.7		Conclusion	33
Chapter 4		Drift-free Motion Estimation from Video Images using Phase Correlation and Linear Opti- mization	36
4.1		Motivation	36
4.2		Optimization of Translational, Rotational and Scaling	37
	4.2.1	Rotation Optimization.	37
	4.2.2	Scaling Optimization.	37
	4.2.3	Translation Optimization	37
4.3		Solving Least Square Problem via Distance Matrix	38
	4.3.1	Definition of Distance Matrix	38
	4.3.2	Conversion to the Least Squares Problem	39
	4.3.3	Normal Equation Derivation Using Distance Matrix's Information	41
		$\mathbf{A}^\top \mathbf{W} \mathbf{A}$ calculation.	41
		$\mathbf{A}^\top \mathbf{W} \mathbf{y}$ calculation.	41
4.4		Experiment and Evaluation	42
	4.4.1	Decision of Weight $\mathbf{\Omega}$	43
	4.4.2	Qualitative Evaluation	44
	4.4.3	Quantitative Evaluation	45
	4.4.4	Computation Time	45
	4.4.5	Comparison between feature point based method	45
4.5		Conclusion	45
Part II Position-Based Method			47
Chapter 5		Ground Vehicle Position Estimation by Monocular Vision and Height Constraints	48
5.1		Motivation	48
	5.1.1	Setup of experimental field	48
	5.1.2	Related works and backgrounds	48
5.2		Robust marker tracking method	49
	5.2.1	Marker used in this paper	49
	5.2.2	Coarse and Fine tracking system	50
5.3		Position estimation via height constraint	51
	5.3.1	Camera projection model	51

	5.3.2	Solve position estimation via height constraint	52
	5.3.3	3D reconstruction from multiple view	52
5.4		Experimental evaluation	53
5.5		Conclusion	54
Chapter 6		Switching Observer based Motion Estimation via Stereo Disparity and Monocular Scaling Sensor Fusion	57
6.1		Motivation	57
	6.1.1	Stereo measurement	57
	6.1.2	Monocular measurement	58
6.2		Algorithm flow	58
	6.2.1	Template extraction based on machine learning	58
	6.2.2	Template tracking using SURF [2] feature and scale extraction	59
	6.2.3	Stereo disparity measurement	59
6.3		Sensor fusion filtering design for stereo and monocular sensing mixture	61
	6.3.1	State-space model	61
	6.3.2	EKF estimation flow	62
	6.3.3	Convergence analysis with linearized poles	63
	6.3.4	Linearized observer based estimation method	63
6.4		Simulation	64
6.5		Experiment	65
	6.5.1	Experimental setup	65
	6.5.2	Results and discussion	65
6.6		Conclusion	70
Chapter 7		Sensor Fusion Tuning toward Vision based Relative Position Control	71
7.1		Motivation	71
7.2		Problem statement in ACC	71
	7.2.1	String stability in ACC	72
	7.2.2	Vision based ACC controller design	73
7.3		Feedback control simulation with state-feedback controller	74
7.4		Conclusion	77
Chapter 8		Observer Designs in Terms of Gaussian Sensor Noise and Convergence Speeds	81
8.1		Motivation	81
	8.1.1	Related work	81
	8.1.2	Proposed approach	82
8.2		Fixed gain observer considering observation error	82
	8.2.1	Error and covariance study on discrete fixed gain observer	82
	8.2.2	Simulative validation for steady-state covariance estimation	83
	8.2.3	Study on sensor choice problem	84
8.3		Sensor noise effect evaluation with state feedback and observations	84
	8.3.1	Observation noise effect to the estimation and controlled error	86

8.3.2	Observation and control separation in noise propagation	87
8.3.3	Practical observer design example with the linear stage parameters	87
8.3.4	Sensor choice considering effects of noise and its application for image processing algorithm evaluation	93
8.4	Conclusion	95
Chapter 9	Conclusion	97
Appendix		100
Appendix A	Homography based planer estimation	100
A.1	Camera projection model to homography [3]	100
A.2	Scaling and rotation extraction	101
Appendix B	3D Rotaion and Homography Transformation	102
Appendix C	Relationship with H_2 norm minimization problem	103
C.1	H_2 norm definition	103
C.2	LMI based H_2 optimization	103
C.3	Analogy between polar constrained variance minimization estimation and H_2 minimization	104
C.4	Matrix inequality based Minimum-estimation-covariance observer design under convergence speed constraints	104
Appendix D	Estimation error covariance minimization via adaptive observer	106
D.1	Error and covariance study on discrete adaptive observer	106
D.2	Study on adaptive gain determination with constraints on convergence	107
Bibliography		108
References		108
List of Publications		114

List of Figures

1.1	Camera VS LIDAR comparison.	2
1.2	Structure of this paper.	3
2.1	The FFT based image registration algorithm flow [4,5]	8
2.2	Appearance of POC function r when detection is (a) succeeded or (b) failed. Existence of the sharp peak means successful estimation.	8
2.3	The actual process flow of the proposed registration method. The yellow boxes contains the tricks described in this paper.	11
2.4	An example of the correlation peaks for the rotation and scaling estimation between $F_{LP}(l_1, l_2)$ and $G_{LP}(l_1, l_2)$. Two peaks shows there are an ambiguity in rotation detection.	14
2.5	The band-pass filter appearance in the whitened magnitude \mathcal{M}_F (left) and Log-Polared Image \mathcal{LP}_F (right).	14
2.6	Reference images used for the evaluation.	15
2.7	Contrast changed images by Eq.2.14 used for the evaluation.	15
2.8	One of the image matching example with the proposed method, while other methods can not achieve registration.	22
3.1	The process to extract image feature reference $\xi_{ref}(t)$ from video frames and feature reference got from this process	26
3.2	Block diagram of the proposed method.	27
3.3	Tracking result using approximated jacobian in Eq. (3.4)	30
3.4	Tracking result using proposed time-invariant jacobian in Eq. (3.5)	31
3.5	Experimental results of distance estimation. Blue line: method using scaling, red line: method using translation	31
3.6	Experimental setup	32
3.7	Raw time derivative of renference $\frac{d}{dt}\xi_{ref}(t)$ in Fig. 3.1.	33
3.8	Smoothed reference $\frac{d}{dt}\xi_{ref}(t)$ by filtering Fig. 3.7	33
3.9	Experimental results without feedforward control. Image feature reference: broken line. Actual feature trajectories: solid line.	34
3.10	Experimental results with proposed feedforward control. Image feature reference: broken line. Actual feature trajectories: solid line.	34
4.1	An example of a distance matrix \mathbf{D} shown in Eq. (4.9)	39
4.2	How to calculate $\mathbf{A}^\top \mathbf{W} \mathbf{A}$	42

4.3	How to calculate $\mathbf{A}^T \mathbf{W} \mathbf{y}$	42
4.4	The way to make a weight matrix in Fig. 4.4(b) from a reliability matrix in Fig. 4.4(a). .	43
4.5	Qualitative evaluation. (a) ground truth image used in evaluation. (b)(c)(d)(e) Optimized transformation parameter. Red broken line shows iterative method and blue line shows proposed method using distance matrix. (f) Mosaiced image with conventional method. (g) Mosaiced image with proposed method. It is obvious that the (g) is mosaiced better than (f) compared with the ground truth	44
4.6	Quantitative evaluation. (a) RMS error between ground truth and each estimated frame. (b) Mutual Information between ground truth and estimated frame. Blue lines, which represent proposed method are better than conventional one represented as red broken lines.	46
4.7	The weight matrix with SIFT and optimization results from SIFT measurement.	46
5.1	Experimental field captured from the ceil camera.	49
5.2	Program flow of the proposed position estimation method.	49
5.3	Course and Fine tracker algorithm flow	50
5.4	Course and Fine tracking comparison.	51
5.5	Depth image of the experimental fields. Lighter area means further distance.	52
5.6	Comparison of the methods used in evaluation.	53
5.7	Evaluation setup. A linear stage with a red marker set along to X axis. The linear encoder was used for groundtruth, then images for estimation were captured from a stereo camera. .	54
5.8	Evaluation with X axis motion.	55
5.9	Evaluation with Y axis motion.	56
6.1	The principle of a stereo camera.	58
6.2	Depth estimation results from each raw measurements.	59
6.3	Simulation Results with Sin Wave Motion.	60
6.4	Linearized pole based analysis. The poles are shown in s plane.	64
6.5	Experimental Setup	66
6.6	Detection with YOLO	66
6.7	Position data from linear encoder, stereo disparity and scaling. (with a sin wave motion)	67
6.8	Position data from linear encoder, stereo disparity and scaling. (with a sudden motion) .	67
6.9	Experiment results with a sin wave motion.	68
6.10	Experiment results with a sudden motion.	69
7.1	The concept of ACC and variable declaration.	71
7.2	The block diagram of the ACC system.	72
7.3	Pole positions of the continuous state-feedback controller of ACC with Eq. (7.8) parameters shown Table. 7.1.	73
7.4	EKF-estimation-based ACC results with the controller class C.	75
7.5	Pole-placement-observer-estimation-based ACC results with the controller class C.	76
7.6	Relative position tracking results comparison with noise and the controller class C. . . .	77
7.7	Velocity tracking results comparison with noise and the controller class C.	77

7.8	EKF-estimation-based ACC results with sensing noise and the controller class C.	78
7.9	Pole-placement-observer-estimation-based ACC results with sensing noise and the controller class C.	79
7.10	Relative position tracking results comparison with noise and the controller class C.	80
7.11	Velocity tracking results comparison with noise and the controller class C.	80
8.1	State observation results.	84
8.2	Discrete state feedback and observation system considering sensor and system noise.	85
8.3	Bode Plots for the linear Stage.	88
8.4	Butterworth poles for control and observation.	89
8.5	Observer noise evaluation via $trace(P_{obs})$ and state noise evaluation via $trace(P_{obs})$ with the butterworth pattern.	89
8.6	Observer noise prediction for each butterworth observation poles and simulation results.	90
8.7	State noise prediction for each butterworth observation poles and simulation results	90
8.8	Overlapping poles for control and observation.	91
8.9	Observer noise evaluation via $trace(P_{obs})$ and state noise evaluation via $trace(P_{obs})$ with the overlapping pole pattern.	91
8.10	Observer noise prediction for each overlapping pole-based observation poles and simulation results.	92
8.11	State noise prediction for each overlapping pole-based observation poles and simulation results	92
8.12	Log scale steady-state variance of each state variable with controller group A.	95
8.13	Log scale steady-state variance of each state variable with controller group B.	95
8.14	Log scale steady-state variance of each state variable with controller group C.	95
C.1	LMI condition for convergence speed for continuous model and discrete model.	105

List of Tables

2.1	Comparison with proposed method and other method without particular processing in Fig. 2.3.	16
2.2	Comparison on processed images with proposed method and other method without particular processing in Fig. 2.3. Success rate for the each estimation method and its average error within each parameter.	17
2.3	Comparison between FFT based method and feature point based method.	19
2.4	Comparison between FFT based method and feature point based method. Success rate for the each estimation method and its average error within each parameter.	20
3.1	Variables in a simulation.	28
4.1	Parameters in evaluation	43
4.2	Average Computation Time	45
6.1	Camera and motion parameter in simulation	62
6.2	EKF parameter in simulation	62
7.1	Feedback gain at each categories [6].	72
8.1	The linear stage parameter	88
8.2	Error analisys of run-time performance of our system on the KITTI (1241 x 376 px, 0.46 MP) dataset [7] in [8].	93
8.3	Sensing noise assumption to calculate each matching method's variance and run-time from [8].	93
8.4	Sensing method comparison result.	94
9.1	Relationships of each chapter.	98

Chapter 1

Introduction

1.1 Background

Image sensors such as a camera can acquire a huge amount of information without touching the object. As shown in Fig. 1.1, a camera has great advantages in cost, portability, and the amount of information. Therefore it has been widely used as an environmental sensor for industrial machines and mobile robots [3,9].

Due to the improvement in the silicon semiconductors processing in recent years, image sensors are becoming smaller, lighter and cheaper with higher frame rate and resolutions. In addition, image processing technology has been developing at an innovative rate since the early 2000s, and it is expected that the possibilities will continue to expand in the future. Therefore, the vision sensor, such as a camera, will play very important roles in robotics automation.

This paper focuses on vision-based control techniques called visual servo, which moves robots and manipulators by feeding back visual information obtained from a camera often mounted on a robot.

Because visual servo theory seamlessly connects sensing, processing, and control strategy, it has been applied in various fields such as automation of operation of industrial machines and control of mobile robots [10–12].

When controlling a robot based on an image acquired from an image sensor, it is necessary to extract some value from the information obtained from the image. An image is treated as a matrix of minimum elements called pixels and has a high degree of freedom. Still, it is necessary to extract some notable values for use in control appropriately. The process of extracting some meaningful value from an image that is a matrix of pixel values is called feature extraction, and the obtained quantity is called a feature [9] [3].

In the visual servo, a command value for the robot motion is generated based on the feature amount extracted from the image. That is, there are two processes in visual servo that extract features from an image and control them using them. Therefore, control results significantly affected by the features chosen.

From the viewpoint of control in conventional visual servo research, there are two types of features: position-based methods and image-based methods. The former method uses three-dimensional position information using stereo vision or the like as a feature amount, and the latter approach uses two-dimensional information such as displacement in an image as a feature amount.

The position-based method can be controlled intuitively by directly using the 3D position but has a problem that it is easily affected by errors in 3D reconstruction due to calibration and numerical errors.

	Resolution	Cost	Reliability	Consumption Energy
Camera	Very High	Low	High	Very Low
LiDAR	High	High	Very High	High

LiDAR	Stereo Camera
\$~400	\$~300
10Hz	30-90Hz
400 points	1280 x 720 pix
0.15 - 12m	0.2-10m

Figure 1.1 Camera VS LIDAR comparison.

On the other hand, the image-based method is controlled based on 2D information such as feature points in the image, so positioning is possible regardless of 3D reconstruction error, while inappropriate trajectory is generated in 3D space [13].

A 2-1 / 2D visual servo that combines the features of both methods has also been proposed [14].

These features selections highly characterize the total control systems, so that we have to select features and control method carefully. It is also necessary to take into consideration various performances of sensing-processing dynamics such as computation time, robustness and accuracy.

Thus, image sensing, processing, and control are highly combined problems so that integrated design guidelines are needed.

In this paper, we start with image-based visual servo considering machine tools and unmanned flying robots with a movable stage as a specific application, portable follow-up microscope [15]. The former part introduces a robust image processing technique to get better control characteristics and its application to the video-based feedforward control, overcoming depth estimation problem [16]. The reference optimization is also conducted to get a more accurate reference.

The second part of this paper is focusing on the position-based visual servo, especially focusing on the depth estimation by sensor fusion or other constraints. Begin with the depth estimation via geometric constraints like [17–19], the later part discussing on the depth measurement such as the fusion of the stereo and monocular observation [20]. The final part discussing a desirable observer design considering both the gaussian-noise and convergence rate.

1.2 Summery of each chapter

Each chapter has its own motivation to solve particular problem: so, instead of explaining each background, this section shows the brief summery of chapters.

First, in chapter 2, we introduce phase-correlation-based image displacement estimation, which extracts geometric displacements, including translation, rotation, and scaling. Because each spacial phase and magnitude information contains the translational and rotation/scaling information, an image displacement can be extracted by performing appropriate filtering, which highly affects accuracy and robustness. As a result of comparing accuracy and robustness with methods using feature points methods, it was shown that there are advantages in usefulness such as specific patterns, calculation time, and false-aware detection.

Next, chapter 3 proposes a method of performing trajectory tracking control by feeding back the two-dimensional displacement detected in chapter 2. In image-based visual servo, it was necessary to calculate

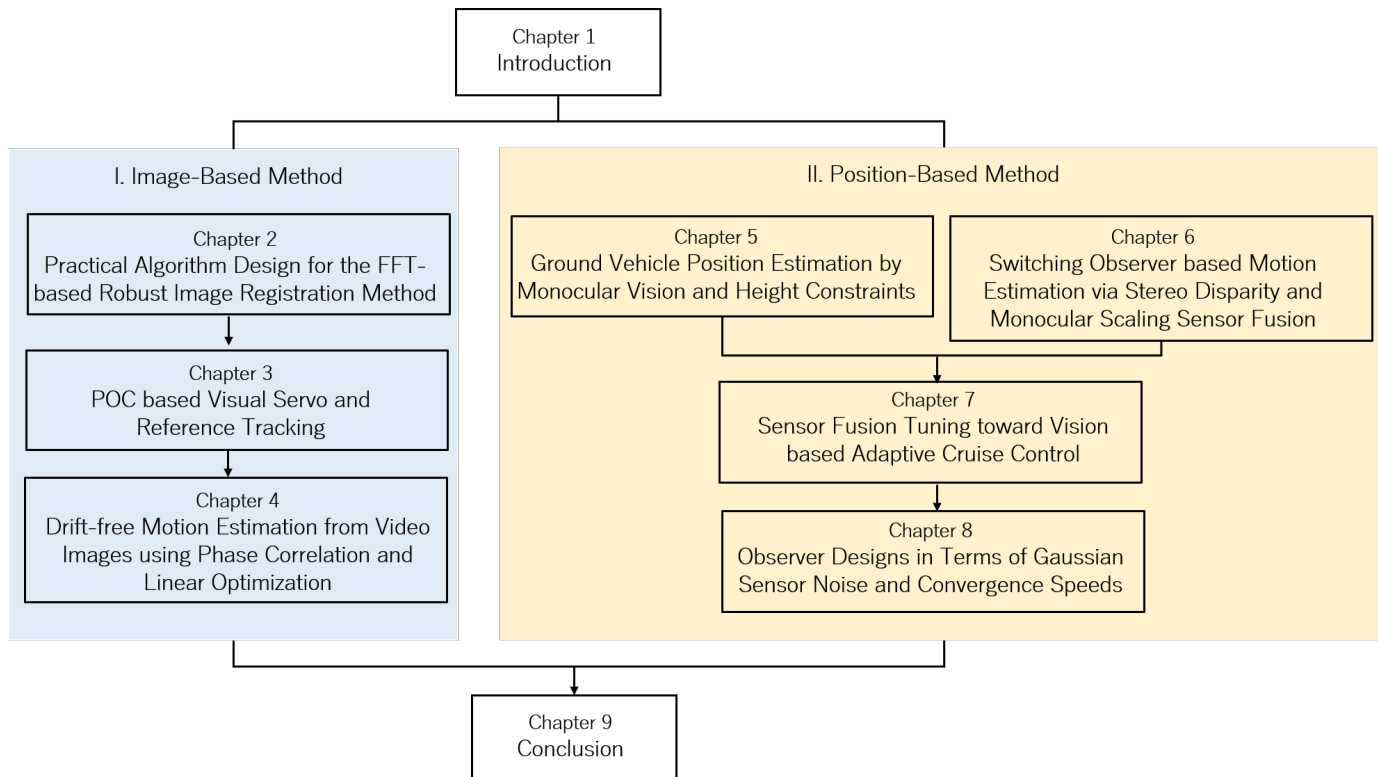


Figure 1.2 Structure of this paper.

the image jacobian that correlates the motion in the image with the motion in three dimensions. By using a control law that matches the sensing, this image jacobian can be converted to a time-invariant matrix. As a result, feedforward control, which was originally impractical because it had to be calculated based on the ever-changing distance information, can be easily implemented with the proposed constant jacobian. This can be applied to visual tracking control to motion teaching. Then chapter 4 shows the reference extraction method from video images: in other words, it is the camera motion estimation method. In the trajectory tracking control described in chapter 3, the operation of the robot can be reproduced based on the teaching operation defined by the video images. At that time, the command value is generated based on the displacement information between consecutive frames of the moving image, but more precise motion reproduction is possible by optimizing full relationships between whole video frames. By proposing an efficient optimization algorithm based on the distance matrix, an effective method for larger-scale trajectory tracking control is proposed.

Chapter 5, is aiming to construct an indoor positioning system for a small ground vehicle, which uses height constraints of markers on the ground vehicle.

Chapter 6 deals with a distance estimation method using a stereo camera. In a stereo camera, a distance is estimated based on parallax information obtained from an adjacent camera, but there is a problem that parallax cannot be sufficiently obtained when an object is at a closer range, and it has tradeoff with accuracy. Therefore, we propose a method for robust distance estimation by combining stereo disparity and the scale change of an object obtained with a single eye observation. In addition, we confirmed that applying the switching mechanism in the extended kalman filter and the linearized observer enables faster and robust estimation. These estimations are evaluated in chapter 7. The estimated depth and velocity are applied to the Adaptive Cruise Control (ACC) system. We try to evaluate estimation algorithms with vision-based ACC.

Finally, chapter 8 considers and proposes a method of designing an estimator from both aspects of

the sensor error, assuming a gaussian distributed sensor noise and the convergence speed. The observer used in control is often designed based on the frequency characteristics, so we propose the precalculation to predict steady-state variance in advance from the Lyapunov equation. It is shown that by using this framework, it is possible to select a controller, sensing algorithm, and estimator filter that satisfies the desired convergence speed and steady-state variance. Further discussion with optimization using matrix inequality or adaptive observer is shown in appendix C and D. And whole frameworks are summarized in chapter 9.

Part I

Image-Based Method

Chapter 2

Practical Algorithm Design for the FFT-based Robust Image Registration Method

2.1 Motivation

In this chapter, we discuss the image registration technique based on the assumption that the object to be observed by the camera is rigid and flat. The image registration technology used at present can be roughly classified into three techniques of optimizing the displacement by the evaluation function based on the pixel, the technique using the local feature quantity, and the technique focusing on the frequency region using the discrete Fourier transform.

In the method based on the evaluation function such as the Lucas-Kanade method [21], the evaluation function such as the sum of residual squares is set after assuming the model of the displacement between images, and the parameters of the displacement for the best evaluation are optimized by the sequential method such as the gradient method. While it has an advantage of [22] that robust parameters can be estimated for illumination fluctuation, etc. by setting an appropriate evaluation function, it has structural problems such as increase in calculation cost by iterative calculation and falling into local solution in optimization.

On the other hand, as local features, feature point detection based on Scale Invariant Feature Transform (SIFT) features [23] and Speed Up Robust Features (SURF) features [2], which are often used in the field of robotics. There is a matching method. Since local correspondence can be obtained, it can be applied to non-planar objects, etc., and it is used for 3D reconstruction, etc. On the other hand, it requires high calculation cost for feature point detection and matching, and incorrect matching. There is also the danger of producing unintentional detection results based on this. [22] In this paper, we propose a registration method based on Fast Fourier Transform (FFT). The method using two-dimensional Fourier transform of an image is well known as Phase Correlation [24], and the translation and rotation of the image and the amount of scale displacement can be determined from the properties of the translation and the constant multiplication of the Fourier transform. [4, 5] that can be detected.

When estimating similar parameters, there are the following three advantages over other methods that can use phase correlation.

Efficiency of calculation Unlike the method that requires iterative calculation such as Lucas-Kanade, this method can derive an estimated value within a predetermined process. Among them, FFT and partial coordinate transformation, which account for the majority of computation, are easy to

perform parallel computation on hardware such as FPGA and GPU and can speed up to the order of a few ms. [25].

Consistency of calculation time Since the process used for calculation is fixed, the calculation time is expected to be constant. The method using local features varies depending on the number of textures and feature points of the image, and the method such as RANdom SAmple Consensus (RANSAC) [26] based on random sampling used for outlier removal. However, there are cases where it becomes even more difficult to make the calculation time constant. [27].

Ease of evaluation By detecting the peak value of the correlation obtained by using the method using phase correlation, it is possible to detect false detections, thus avoiding the risk of malfunction in control systems.

An efficient calculation can be implemented relatively easily, and the nature of not involving processes such as optimization and matching search is suitable for applications that require real-time performance such as visual servo driving a robot based on image information. It also has the advantage that dangerous robot motion can be prevented by actively eliminating unintended registration results by evaluating the similarity obtained during displacement detection.

Although the image registration technique based on phase correlation has these advantages, it can be found in the literature [4] when estimating four-degree-of-freedom parameters including rotation and scaling in addition to a translation. There is a problem that sufficient robustness cannot be ensured by the method based on public information. Aoki et al. [5, 28] later provided some guidelines such as appropriate frequency filtering and whitening, but algorithm design still requires many manual tunings.

In this chapter, we aim to improve the robustness and accuracy of the image registration method based on FFT and clarify the design guidelines such as tuning the hyperparameters of each filter together with their geometric meanings.

2.2 The principle of displacement estimation via Phase Correlation

Displacement estimation using phase correlation can be derived from the characteristics of Fourier transform. Fig. 2.1 shows the basic phase correlation algorithm flow.

2.2.1 Translation Estimation

Let two images f and g have a translational displacement (x_0, y_0) i.e.,

$$g(x, y) = f(x - x_0, y - y_0) \quad (2.1)$$

The 2D Fourier transforms of Eq. (2.1) becomes follows:

$$G(\chi, \psi) = e^{-j2\pi(\chi x_0 + \psi y_0)} F(\chi, \psi) \quad (2.2)$$

where F and G means 2D Fourier transformation of f and g . By taking cross-power spectrum of F and G to its magnitude, we get phase part R .

$$R(\chi, \psi) = \frac{FG^*}{|FG^*|} = e^{j2\pi(\chi x_0 + \psi y_0)} \quad (2.3)$$

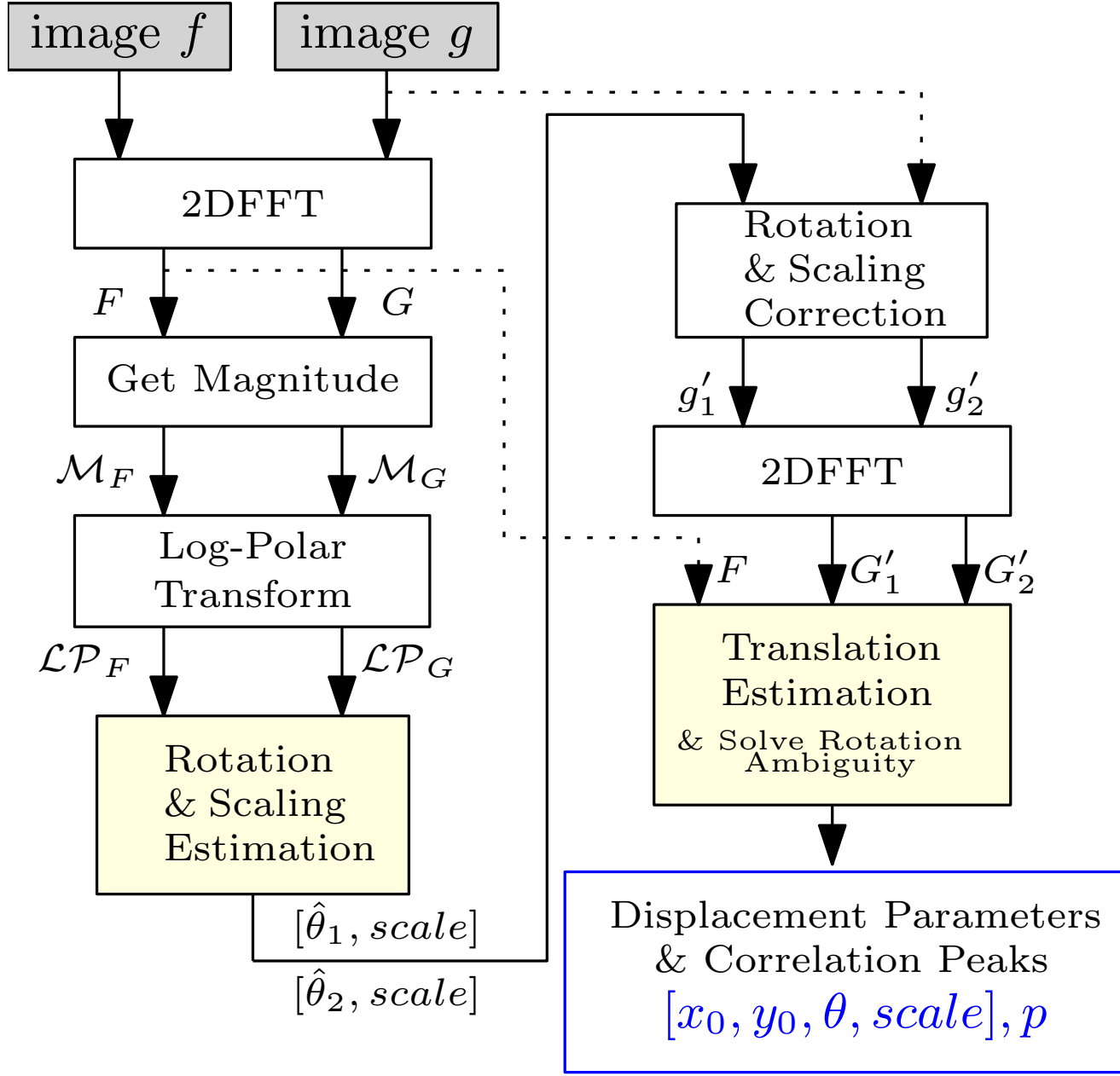


Figure 2.1 The FFT based image registration algorithm flow [4, 5] .

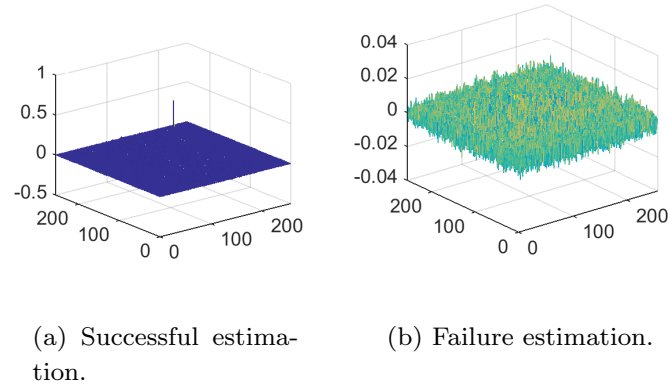


Figure 2.2 Appearance of POC function r when detection is (a) succeeded or (b) failed. Existence of the sharp peak means successful estimation.

The inverse Fourier transform of R becomes a Dirac δ -distribution centered in (x_0, y_0) .

For the real case, r is distorted to be blunted peak due to window functions [5].

2.2.2 Rotation and Scaling Estimation

Considering the case that image f and g have rotational displacement θ_0 and scaling κ around image center, this relationship can be written as following equations:

$$g(x, y) = f(x', y'). \quad (2.4)$$

$$x' = \kappa \cos \theta_0 x - \kappa \sin \theta_0 y + x_0 \quad (2.5)$$

$$y' = \kappa \sin \theta_0 x + \kappa \cos \theta_0 y + y_0 \quad (2.6)$$

while $f(x, y)$ means pixel intensity at coordinate (x, y) of image f .

By applying the Fourier transform constant multiple theorem to Eq. (2.4), the rotation, scaling and translation components can be separated like Eq. (2.7).

$$G(\chi, \psi) = e^{-j2\pi(\chi x_0 + \psi y_0)} F(\kappa \cos \theta_0 \chi - \kappa \sin \theta_0 \psi, \kappa \sin \theta_0 \chi + \kappa \cos \theta_0 \psi) \quad (2.7)$$

From Eq. (2.7), only the rotation and scaling information is stored in the amplitudes of the Fourier transforms F and G of the image f, g , so the amplitudes of F and G . The relationship between $\mathcal{M}_F, \mathcal{M}_G$ is as shown in Eq. (2.8).

$$\mathcal{M}_G(\chi, \psi) = \mathcal{M}_F(\kappa \cos \theta_0 \chi - \kappa \sin \theta_0 \psi, \kappa \sin \theta_0 \chi + \kappa \cos \theta_0 \psi) \quad (2.8)$$

By converting these amplitudes $\mathcal{M}_F, \mathcal{M}_G$ into log-polar coordinate (Log-Polar), rotation and scale displacement can be converted into translational displacement. If the origin of the image is (cx, cy) , the image $\mathcal{LP}_F(\theta, \rho)$ and the original amplitude \mathcal{M}_F . The relationship is expressed as follows:

$$\mathcal{LP}_F(\theta, \rho) = \mathcal{M}_F(cx + e^{\frac{\rho}{M}} \cos \theta, cy + e^{\frac{\rho}{M}} \sin \theta). \quad (2.9)$$

Here, M is a constant that determines the degree of compression of information in log-polar coordinate transformation. In this log-polar transformed image, the rotation and scaling of the original image are expressed as translational displacements in the orthogonal axes θ, ρ direction after transformation respectively.

By using the rotation amount and scaling amount obtained after that, the image g_{rev} is generated by correcting the rotation and scaling of the image g . Since only translational displacement exists between g_{rev} and f , the final displacement including translation can be obtained using the translation detection method.

2.3 Proposed Algorithm

The detection principle of image displacement based on FFT is shown in Chapter 2, but there are some important items to ensure the robustness necessary for application to an actual system [28]. The entire algorithm flow including 2.3.4 filtering and 2.3.6 sub-pixel estimation processing newly proposed in this paper is like Fig. 2.3. By clarifying the significance of all these processes, we propose an algorithm that

allows users to tune flexibly according to the situation.

2.3.1 Window Function

The robustness of the detection can be remarkably improved by reducing the effect of the screen edge using the window function in the two-dimensional FFT. Since it is known that the shape of the peak value changes by the selection of the window function, the selection of the window function can also be a tuning object, but since the significant difference could not be verified, this paper adopts the following two-dimensional thinning window $w(n)$ used in many other literatures.

$$w(n) = \frac{1}{4} \left(1 + \cos\left(\pi \frac{y}{H}\right) \right) \left(1 + \cos\left(\pi \frac{x}{W}\right) \right) \quad (2.10)$$

while, H and W mean image height and width.

2.3.2 Whitening

When calculating the amplitude \mathcal{M}_F in Eq. (2.8), the amplitude of the low frequency is exponentially larger than that of the high frequency, so we apply whitening process to make use of the higher frequency information. Following Aoki [28], the following whitening process is applied.

$$\mathcal{M}_F(\chi, \psi) = \log(|F(\chi, \psi)| + 1) \quad (2.11)$$

2.3.3 Scaling Parameter Desition for Log-polar Transformation

In the log-polar coordinate transformation of Eq. (2.9), the determination of the constant M is a very important factor directly related to accuracy and robustness.

The reason why the constant M is necessary is that the log-polar coordinate conversion compresses pixels on the circumference of the radius r to a small value of $\log r$ on the converted linear scale, and the converted image becomes extremely smaller.

The amount of information after coordinate transformation can be maintained by setting M to a certain extent, but it does not lead to a significant improvement in accuracy due to the lack of information due to linear interpolation. Since the pixels at the corners, which are $\frac{N}{\sqrt{2}}$ away from the center of the original image \mathcal{M}_F , are projected onto $M \log \frac{N}{\sqrt{2}}$ at ρ axis, the M should be designed so that the whole pixels are fit within the screen after conversion. And that condition is written as $M \log \frac{N}{\sqrt{2}} < N$.

As a result of trial and error, we used $M = \frac{N}{\log N}$ which is also used in [5].

With the our design, the relationship between rotational and scaling displacement can be written as $(\Delta\theta, \Delta\rho) = (\theta_0 \frac{N}{2\pi}, M \log \kappa) = (\theta_0 \frac{N}{2\pi}, N \log_N \kappa)$.

2.3.4 Bandpass Filtering for Rotation-Scaling Estimation

Similar to 2.3.3 design, bandpass filtering at frequency domain plays an important role in the rotation and scaling estimation. We propose the design of this bandpass cutoff value from geometrical considerations.

Past research [5] and [29] used a log-polar coordinate transformation, which works as a high-pass filter,

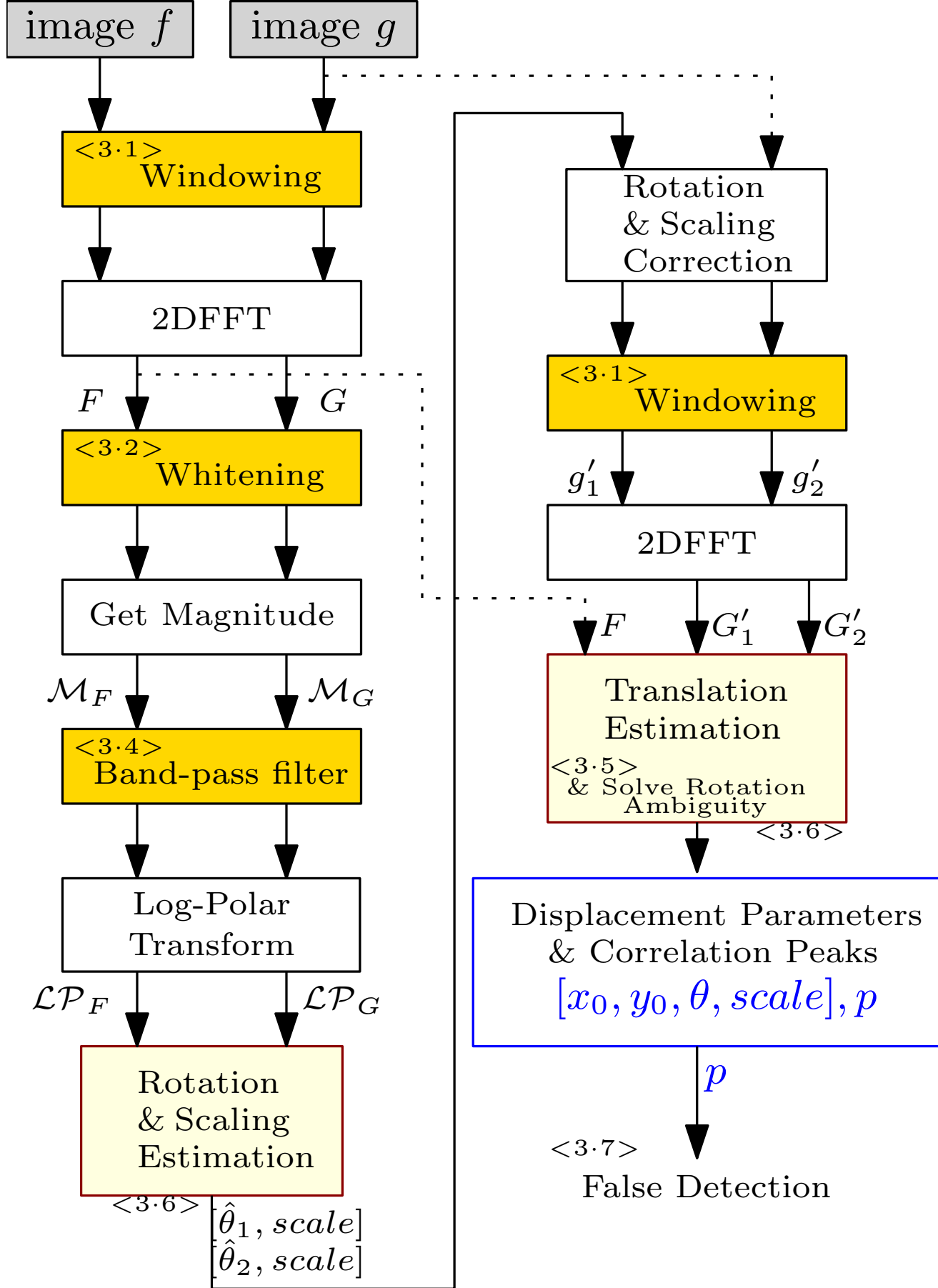


Figure 2.3 The actual process flow of the proposed registration method. The yellow boxes contains the tricks described in this paper.

to the amplitude M_F, M_G to improve the accuracy.

As a geometrical meaning, when a log-polar transformation is performed as shown in Fig. 2.5, the estimated $2\pi r$ pixel information existing on the circumference of radius r To correspond to N pixel information, This is because the low-frequency information near the center before conversion is greatly enlarged and becomes a small amount of information. On the other hand, it is known that the signal-to-noise ratio is low in the high-frequency region corresponding to the screen edge before conversion, so it is necessary to apply some low-pass to obtain robustness against noise. Therefore, it is necessary to design a band-pass filter that appropriately cuts out the low-frequency and high-frequency information. Since the cutoff value of this filter varies depending on the size of the image, we need general design guidelines for tuning the filter from the geometrical meaning.

In the design of the high-pass filter in this paper, the ratio α of the number of pixels $2\pi r$ on the circumference before conversion to the number of pixels N after conversion is the tuning parameter r_{min} Is calculated from Eq. (2.12).

$$2\pi r_{min} = \alpha N \quad (2.12)$$

When α is 1, the pre-conversion pixel ratio and post-conversion pixel ratio are equivalent, and the lower the α is designed, the lower frequency information will be taken.

On the other hand, the design of a low-pass filter that cuts off the rightmost part after conversion in Fig. 2.5 is designed using the tuning parameter β as follows:

$$r_{max} = \beta \frac{N}{2} \quad (2.13)$$

When β is 1 in this equation, this low-pass filter cuts the area outside the circle inscribed in the image before conversion, and the lower the β , the lower the low-pass bandwidth.

Using the relation $\rho = M \log r$, we get the the filter cutoff after coordinate conversion is $\rho_{min} = M \log \frac{\alpha N}{2\pi}$ and $\rho_{max} = M \log \beta \frac{N}{2}$.

As a result of trial and error, we decided that $\alpha = 0.5, \beta = 0.8$ are appropriate for $N = 256$. Smaller α and β results in discarding higher frequency information, so it improves robustness for some image noise but gets a less accurate estimation.

2.3.5 Refinement of Rotation and Scaling Estimation

Rotation and scaling information are included in the amplitude of the frequency domain image. Since amplitude in the frequency domain has a pattern with origin symmetry, we can get two candidates for the rotation angle value estimation. Fig. 2.4 shows the peaks representing its ambiguity in the actual estimation process.

Therefore, as the flow shown on the right side of Fig. 2.3, the rotation and scaling of the image g are corrected using the obtained two rotations and scalings to g'_1 and g'_2 . Then translation estimations are applied for each corrected image and we get 2.2, in which the correct rotation angle is discriminated based on the magnitude of the two detected phase correlation peak values.

2.3.6 Subpixel Matching

The peak of the correlation value of the phase correlation in the non-ideal environment has a shape similar to the sinc function and is discretized in units of pixels. Using some interpolation or fitting technique can perform subpixel accuracy estimation [30]. [5] and [30] models the shape of the peak and performs fitting using observation information, but fitting to this non-linear function requires repeated calculations and is susceptible to noise [31]. Therefore, this paper adopts the centroid value at 5 *times* 5 pixels around the peak value [31], which is the simplest implementation among many subpixel matchings and is considered to have good performance, as the subpixel estimation value.

We also propose to discard smaller value at centroid calculation to get noise suppression and set its threshold ratio toward the peak value as $c = 0.1$.

2.3.7 Estimation Varidation

Finally, we get the displacement parameter $[x_0, y_0, \theta, \kappa]$ and peak value p shown in Fig. 2.2. This peak value becomes $0.1 - -1.0$ in successful estimation and lower in the other case, so we can easily distinguish successful estimation by setting an appropriate threshold p_{th} . Generally, images with clear texture have higher peaks and this threshold design depends on situations. With sampled images shown in Fig. 2.6, we chose $p_{th} = 0.06$.

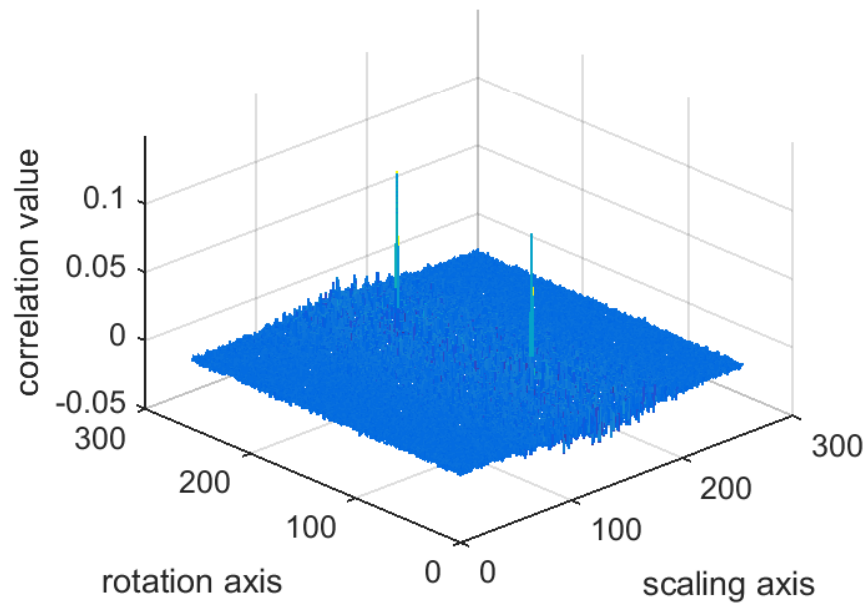


Figure 2.4 An example of the correlation peaks for the rotation and scaling estimation between $F_{LP}(l_1, l_2)$ and $G_{LP}(l_1, l_2)$. Two peaks shows there are an ambiguity in rotation detection.

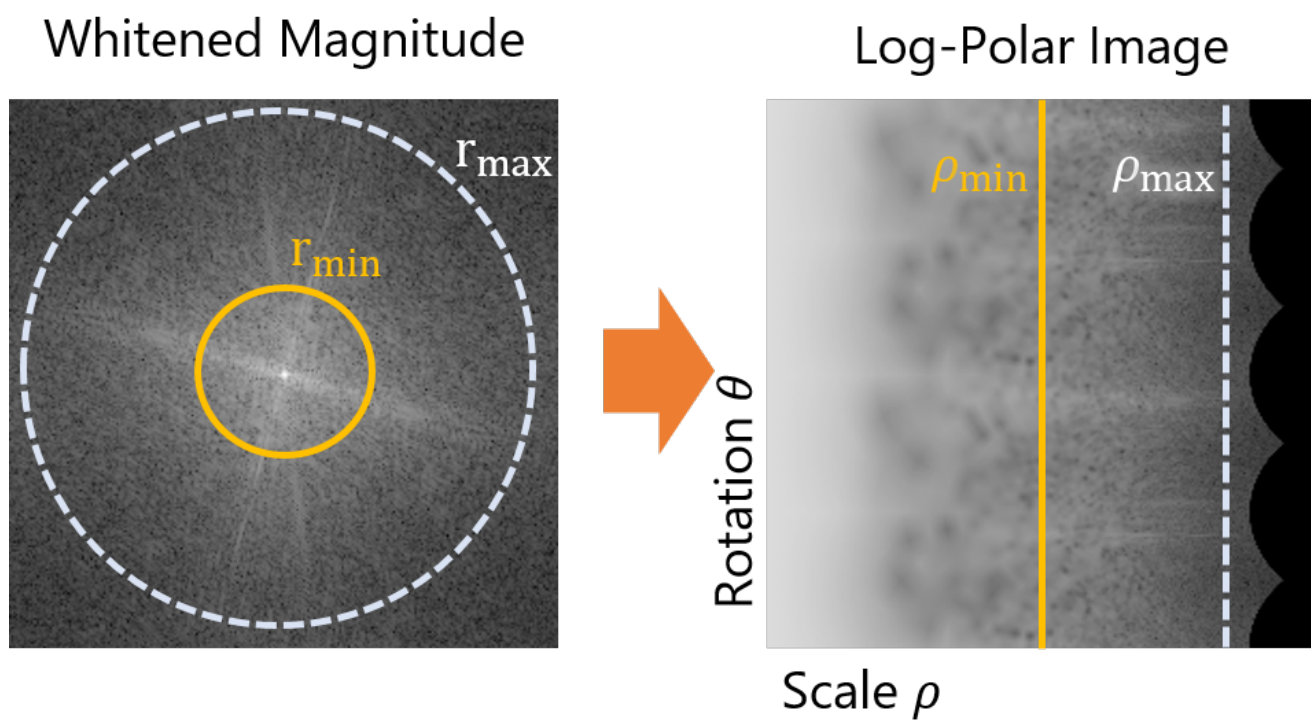
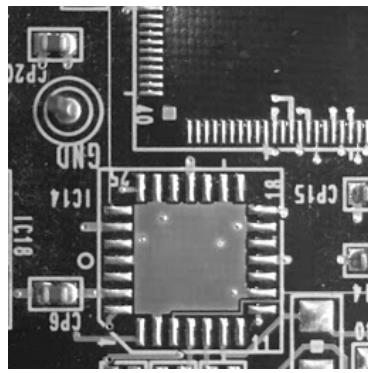


Figure 2.5 The band-pass filter appearance in the whitened magnitude \mathcal{M}_F (left) and Log-Polared Image \mathcal{LP}_F (right).



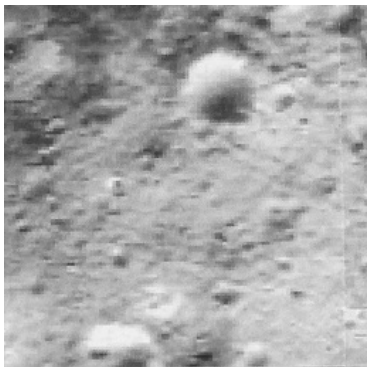
(a) Airmap*¹



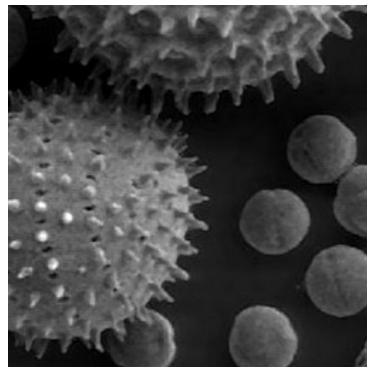
(b) Circuit



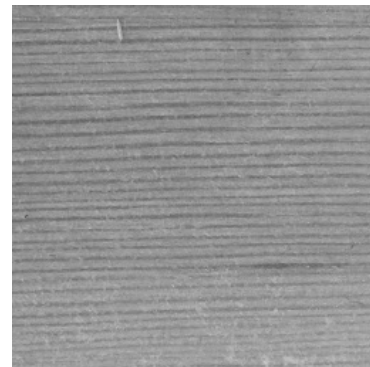
(c) Clouds



(d) Lunar



(e) Pollen

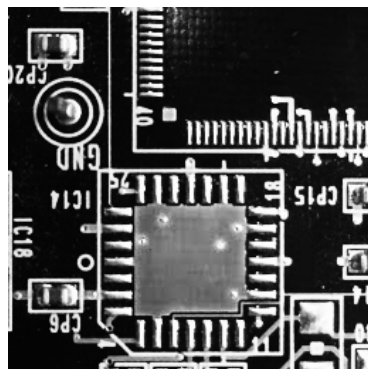


(f) Wood

Figure 2.6 Reference images used for the evaluation.



(a) Airmap



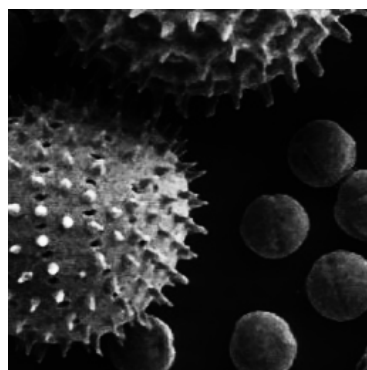
(b) Circuit



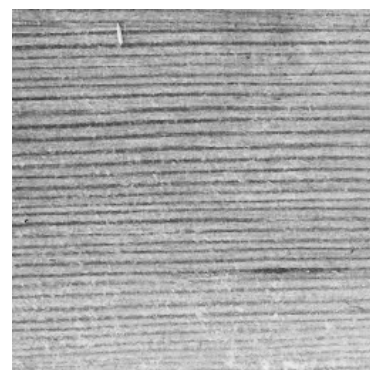
(c) Clouds



(d) Lunar



(e) Pollen



(f) Wood

Figure 2.7 Contrast changed images by Eq.2.14 used for the evaluation.

Table 2.1 Comparison with proposed method and other method without particular processing in Fig. 2.3.

Estimation Results		Comparison of Estimation Error in Same Contrast Situation Matching Rate[%], Average Error $\Delta[x, y, \theta, \kappa]$ (pix,pix,deg,scale)							
Templates		airmap	circuit	clouds	lunar	pollen	wood		
Proposed		0.17	0.092	0.52	0.14	0.17	0.66		
		96%	98%	96%	91%	96%	97%		
		0.14	0.15	0.39	0.16	0.17	0.67		
		0.11	0.11	0.11	0.12	0.075	0.097		
No whitening		0.0067	0.0069	0.0077	0.0067	0.0062	0.0074		
		0.27	0.12	0.53	0.20	0.29	0.68		
		0.20	0.16	0.38	0.22	0.27	0.65		
		0.16	0.14	0.12	0.14	0.11	0.19		
No low-pass		0.010	0.0073	0.010	0.010	0.0053	0.0077		
		0.13	0.033	0.51	0.074	0.091	0.58		
		0.086	0.063	0.35	0.091	0.096	0.64		
		0.080	0.077	0.078	0.071	0.092	0.068		
No Filtered Center of Gravity		0.0054	0.0048	0.0055	0.0055	0.0047	0.0058		
		0.27	0.18	0.58	0.17	0.26	0.66		
		0.26	0.21	0.34	0.25	0.31	0.67		
		0.20	0.25	0.31	0.25	0.29	0.19		
		0.0099	0.0088	0.012	0.010	0.010	0.0096		

Table 2.2 Comparison on processed images with proposed method and other method without particular processing in Fig. 2.3. Success rate for the each estimation method and its average error within each parameter.

Estimation Results	Comparison of Estimation Error in the Different Contrast Situation							
	Matching Rate[%], Average Error $\Delta[x, y, \theta, \kappa]$ (pix,pix,deg,scale)							
Templates	airmap	circuit	clouds	lunar	pollen	wood		
Proposed	0.18	0.095	0.51	0.15	0.24	0.67		
	86%	89%	94%	79%	68%	94%		
	0.16	0.15	0.39	0.17	0.19	0.67		
	0.13	0.13	0.097	0.18	0.16	0.12		
No whitening	0.0074	0.0074	0.0078	0.0078	0.0059	0.0077		
	0.26	0.11	0.55	0.18	0.38	0.68		
	0.20	0.17	0.37	0.20	0.29	0.66		
	0.14	0.14	0.10	0.18	0.16	0.20		
No low-pass	0.010	0.0077	0.010	0.0094	0.010	0.0076		
	0.10	0.035	0.51	0.085	0.11	0.50		
	0.094	0.063	0.30	0.10	0.10	0.63		
	0.080	0.077	0.097	0.16	0.081	0.074		
No Filtered Center of Gravity	0.0059	0.0048	0.0058	0.0090	0.0046	0.0061		
	0.26	0.19	0.57	0.22	0.36	0.67		
	0.25	0.20	0.33	0.18	0.31	0.70		
	0.19	0.24	0.33	0.44	0.39	0.24		
							0.0087	0.010

2.4 Accuracy and Robustness Evaluation

The algorithm was implemented using MATLAB and Python, and the detection performance was verified after creating a data set that recorded the displacement values corresponding to the displaced images. As performance evaluation items, an average estimation error (hereinafter referred to as accuracy) of the translation / rotation / scale amount of the image group that has been successfully detected and a ratio (hereinafter referred to as detection rate) at which the detection has been determined to be successful are used.

Source code using Python and OpenCV used in this method ^{*2} and data set ^{*3} has been released, and this verification has been confirmed to work in the Python 3.5 environment of Windows.

2.4.1 Data set for Evaluation

A dataset for estimating the image displacement parameters were created for the evaluation of the proposed method. After processing the original high-resolution image, 256×256 size reference images shown in the Fig. 2.6 were extracted from the center of original images. Compared images were cropped from transformed window with a randomly generated displacement $\xi_{cmp_i} = [dx_i, dy_i, \theta_i, \kappa_i]$ ($i = 1, 2, 3 \dots N$) .

As shown in Fig. 2.6, our dataset contains six types of images: aerial photograph^{*4}, circuit, clouds, Lunar surface^{*5}, pollen^{*6}, and wood grain^{*7}. These images are chosen considering to contain a wide variety of texture and contrast so that we can use this method in many applications.

Then, the robustness validation was held with lighting changed images in Fig. 2.7. Using intensity conversion

$$i_{new} = \frac{255}{1 + \exp(-\gamma \frac{i-128}{255})} \quad (2.14)$$

, the lighter parts become lighter, the darker parts become darker. We used $\gamma = 15$ at Fig. 2.7.

^{*2} Created displacement estimation library <https://github.com/YoshiRi/ImRegPOC>

^{*3} Created data The set <https://drive.google.com/drive/folders/1bs0N55Xe4KzFZBimSqyq8bSr1KwqwccW>

^{*4} From Google Map

^{*5} From NASA web page <http://wms.lroc.asu.edu/lroc>

^{*6} Pixabay: Free image <https://pixabay.com/ja/photos/pollen/>

^{*7} the rest images are taken with iPhone camera.

Table 2.3 Comparison between FFT based method and feature point based method.

Estimation Results	Comparison of Estimation Error in Same Contrast Situation							
	Matching Rate[%], Average Error $\Delta[x, y, \theta, \kappa]$ (pix,pix,deg,scale)							
Templates	airmap	circuit	clouds	lunar	pollen	wood		
FFT based (Proposed)	0.17	0.092	0.52	0.14	0.17	0.66		
	0.14	0.15	0.39	0.16	0.17	0.67		
	0.11	0.11	0.11	0.12	0.075	0.097	97%	
	0.0067	0.0069	0.0077	0.0067	0.0062	0.0074		
SIFT based	0.35	0.30	0.69	0.32	0.31	0.95		
	0.33	0.31	0.69	0.32	0.34	0.95		
	0.013	0.0084	0.16	0.0096	0.0080	0.038	93%	
	0.00026	0.00012	0.0029	0.00015	0.00012	0.00057		
ORB based	0.97	0.49	NaN	0.66	1.5	0.60		
	0.73	0.407	NaN	0.54	1.0	0.48		
	0.61	3.9	NaN	4.1	1.6	0.31	24%	
	0.017	0.0057	NaN	0.0077	0.019	0.0052		

Table 2.4 Comparison between FFT based method and feature point based method. Success rate for the each estimation method and its average error within each parameter.

Estimation Results		Comparison of Estimation Error in the Different Contrast Situation							
Templates		airmap	circuit	clouds	lunar	pollen	wood		
FFT based (Proposed)		0.18	0.095	0.51	0.15	0.24	0.67		
		0.16	0.15	0.39	0.17	0.19	0.67		
		0.13	0.13	0.097	0.18	0.16	0.12	94%	
		0.0074	0.0074	0.0078	0.0078	0.0059	0.0077		
SIFT based		0.57	0.30	1.3	0.47	0.33	0.95		
		0.78	0.31	0.98	0.38	0.47	0.95		
		0.63	0.025	0.84	0.13	0.14	0.038	100%	93%
		0.019	0.00043	0.017	0.003	0.0030	0.00057		
ORB based		0.13	0.033	NaN	2.7	3.7	0.78		
		0.086	0.063	NaN	1.4	2.2	0.61		
		0.080	0.077	NaN	1.6	10.9	0.28	79%	19%
		0.0054	0.0048	NaN	0.020	0.056	0.0040		

2.4.2 Performance Comparison

Then, we will examine the effectiveness of each proposal explained in the previous section. For comparison, methods without whitening in 2.3.2, high-frequency removal in 2.3.4, and subpixel matching in 2.3.6 are evaluated. A comparison of the accuracy and detection rate was performed for the three cases.

When the window function is not applied to 2.3.1, matching was not achieved for most images, so this item is not compared. Two ratios were used: the rate of success or failure of detection detected based on the proposed threshold and the average error of each displacement.

Although calculation times depends on the PC situation, FFT based and SIFT based method took 50 – 100 ms and ORB based one took 10 – 30ms.

Table. 2.1 shows the detection rate and average error of each method for each image in the dataset. When whitening is not applied, both accuracy and detection rate is reduced compared to the proposed method. On the other hand, if the information in the higher frequency region is not removed, the accuracy is slightly improved, but the detection rate is greatly reduced. Furthermore, the verification by the data set showed that the difference of accuracy is around 1.1 – –2 when the subpixel estimation is ignored or not.

Next, Table. 2.2 also shows the change in the detection rate when the change in brightness is included. Although the detection rate is uniformly reduced, it is shown that the average error at the time of detection is hardly changed, indicating that error detection is being performed correctly. The robustness to lightness is not significantly affected by images that store textures such as Fig. 2.7(c), but for images that have partially lost textures such as the moon surface image of Fig. 2.7(d). The detection rate is greatly reduced. By comparing this fact and the image that was actually detected successfully and the image that failed, it was experimentally confirmed that the registration technique based on FFT depends on the richness of the texture of the image to be compared.

As can be seen from these comparisons, it is clear that each device proposed in image registration using FFT has an effect on robustness and accuracy. Comparing the detection rates of the first and third lines of Table. 2.1 shows that the influence of the design of the bandpass filter of 2.3.4 is particularly large. The appropriate cut-off frequency for the frequency filter in 2.3.4 depends on the size of the image and the image type, but the proposed method provides the intuitive filter design with the geometrical significance of α and β in 2.3.4. In addition, while the newly proposed method of rounding down values below a certain value during subpixel estimation is easy to implement, there is a clear improvement in accuracy, so it is a practical option for subpixel estimation using the center of gravity.

There is a part where the detection rate of the proposed method is inferior to that of the original method in the image of pollen and the circuit board with a strong texture. This is because the lower p_{th} is used, which results in a high detection rate and lower accuracy. This indicates that p_{th} needs to be appropriately tuned by the image.

2.4.3 Comparison with Feature Points-based Method

Next, Table. 2.3 and Table. 2.4 are compared with registration methods using feature points. We selected SIFT [23], which is often used in robotics, and ORB [32], which is frequently used in real-time applications because of its high speed. When estimating the displacement using these feature points,

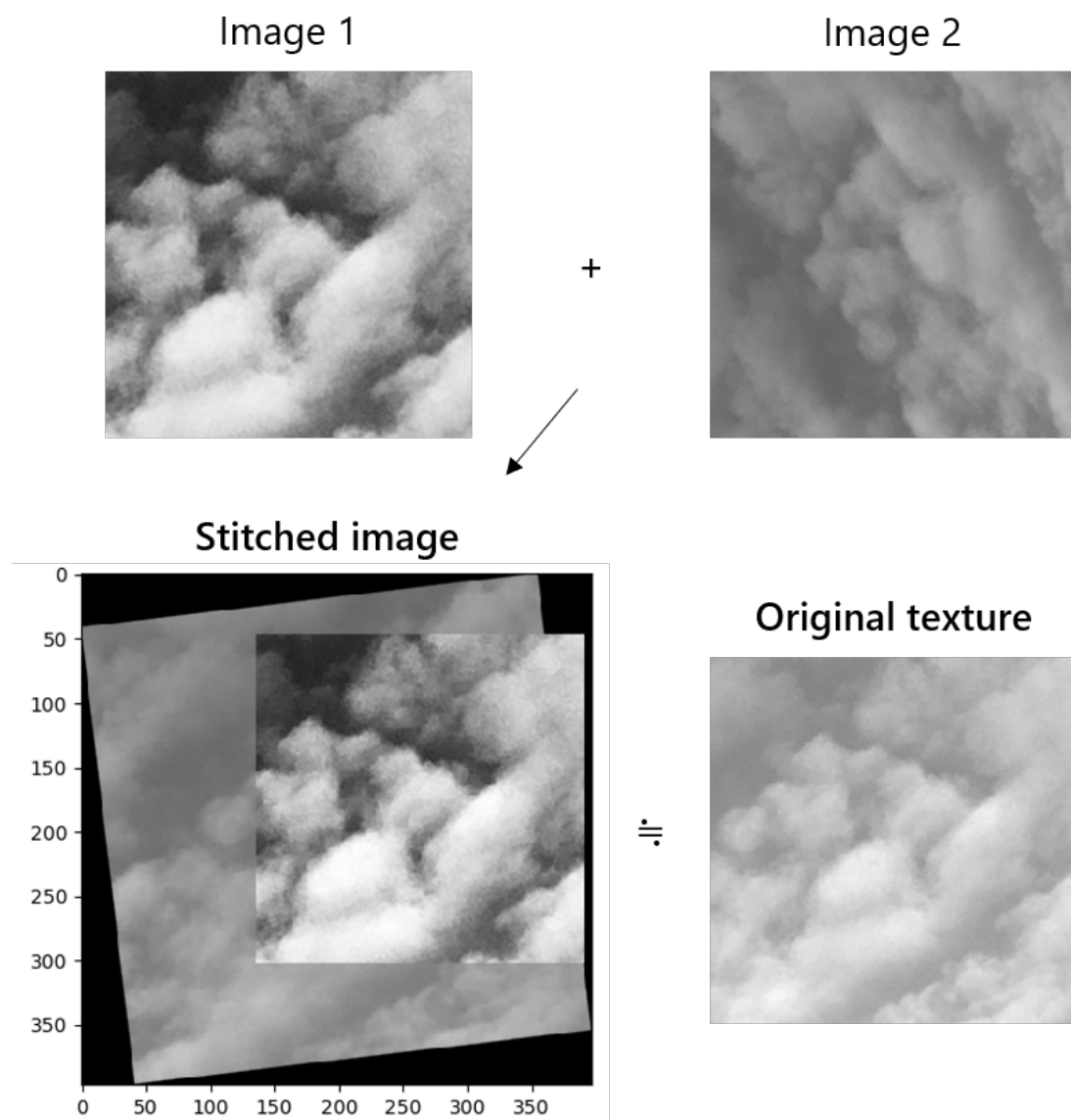


Figure 2.8 One of the image matching example with the proposed method, while other methods can not achieve registration.

the outlier removal method RANSAC [26] based on random sampling is used to reduce the influence of outliers, and the number of remaining corresponding points is four or more. The detection was successful.

The method using ORB can measure displacement quickly and accurately for images with strong textures shown in Fig. 2.6(a) and Fig. 2.6(b), but it is weak against changes in brightness, You can see that they are not good at images with blurred textures such as Fig. 2.6(f) in clouds. Although SIFT has both robustness and accuracy as a whole, sufficient feature points could not be obtained for images with vague textures and results in bad estimation. Fig. 2.8 shows an example of an image in which displacement was obtained only by the proposed method. In the comparison of accuracy, the error of the proposed method is different from the feature-based method, maintaining high accuracy of less than 1 pixel translation and less than 1 degree of rotation, and it can be seen that the data at the time of detection failure can be discriminated well.

For phase correlation, the overlap area of two images to be compared needs to be larger than a certain level in principle, but in the case of feature point-based methods such as SIFT and ORB, if the feature points are abundant, that is, if the texture is abundant, it is narrower There is a possibility of matching between regions, which seems to have led to a difference in detection rate in images of Fig. 2.6(e). However, as shown at the beginning, the phase correlation-based proposed method may be more advantageous in

terms of speedup, hardware implementation advantages, detection of false detections, etc. It is necessary to switch the method flexibly.

2.5 Conclusion

In this paper, we clarified the physical and geometric meaning of each process for the displacement detection method using frequency domain information for image FFT, and applied filter design guidelines and subpixel estimation methods to improve accuracy and robustness. A design guideline for a series of algorithms was studied. For an actual image, it is necessary to change some parameters such as the bandpass filter according to the texture, lighting conditions, and image size of the target, but the proposed algorithm interprets them intuitively.

Furthermore, from the verification based on actual data sets, it was clarified how much each device contributes to accuracy and robustness, and comparison with the methods such as SIFT feature points often used in other fields. The proposed method still has the advantages with vague images like Fig. 2.6(c), and error discrimination based on peak value is switching at the time of motion generation such as visual servo and panoramic image creation [33].

Chapter 3

POC based Visual Servo and Reference Tracking

3.1 Motivation

Visual navigation or path following is one of the attractive theme which contains both recognition process and control process. Image based navigation often uses some geometric feature such as lines [34] or points marker [12] to help recognizing environment. Therefore, robust and accurate geometric feature extraction method such as SIFT [35] or ORB [32] is important for robots to control their motion from image.

On the other hand, some interesting researches use direct approach, which utilize information of image pixels, to determine robot move and achieve path following problem [36] [37]. However, these methods consider the reference path only and do not care timing information.

There are position estimation problem from relative attitude .

In this paper, we assume a plane object and a robot has 4 DOF including 3D translation and rotation around optical axis. This assumption simulates for UAV guidance or robot teaching such as a precise positioning stage.

The goal is to reproduce motion from video; in other words, desired task is to match current image with reference video frame in same time scale. To accomplish this task, control method based on image based visual servo [13] was applied.

Proposed solution has two process: feature extraction process from reference video and tracking control process using visual servo. First, robust feature extraction algorithms based on phase correlation [30] [5] is proposed. Then, the image reference are extracted from video using the proposed algorithms with certain key frame. Finally, tracking control is achieved on proposed effective feedforward visual servo scheme.

3.2 Image Based Visual Servo

Visual servo is a control technique that take some information from a camera image as a feedback and then decide a robot move iteratively. There are roughly two approaches, one is position based visual servo and the other is image based visual servo [13].

In a position based approach, relative camera pose detected from visual measures is used as a control

input. Although an adequate 3D trajectory can be obtained from this approach, vision based pose estimation often has uncertainty and the error in 3D reconstruction directly affect performance. On the other hand, image based approach can suppress this 3D reconstruction error by using an image feature as a control input. Instead, 3D trajectory with image based approach sometimes became undesirable one, this problem can be avoided using combined advanced approach [38]. The control scheme in this paper is based on image based method.

In image based visual servo, we need to define what kind of image feature χ to be used as control input. Once an image feature χ is chosen, reference velocity of robot \mathbf{V}_{ref} can be calculated as below:

$$\mathbf{V}_{ref} = -k_p \mathbf{J}^+ (\chi_{ref} - \chi_c) \quad (3.1)$$

while k_p means a feedback gain, χ_{ref} and χ_c are desired image features and current one. \mathbf{J}^+ means pseudo inverse of \mathbf{J} , which is called image jacobian matrix representing for relationships between infinitesimal displacement of image feature $\dot{\chi}$ and camera velocity \mathbf{V} in Eq. (3.2).

$$\mathbf{J}\mathbf{V} = \dot{\chi} \quad (3.2)$$

generally, \mathbf{J} can be expressed as function of relative pose \mathbf{X} and image feature χ . Since the relative camera pose is time-variant and often hard to estimate from visual measures, it is also difficult to estimate this image jacobian \mathbf{J} in Eq. (3.1).

For example, when using a set of feature points coordinates $\chi_i = [x_i, y_i]^T$ ($i = 1, 2, 3, \dots, n$) as an image feature, χ and camera velocity $\mathbf{V} = [v_x \ v_y \ v_z \ \omega_x \ \omega_y \ \omega_z]^T$ have a following relationship:

$$\dot{\chi}_i = \mathbf{J}_i \mathbf{V}$$

$$\mathbf{J}_i = \begin{bmatrix} \frac{-f}{Z_i} & 0 & \frac{x_i}{Z_i} & \frac{x_i y_i}{f} & -\frac{f + x_i^2}{f} & y_i \\ 0 & \frac{-f}{Z_i} & \frac{y_i}{Z_i} & \frac{f + y_i^2}{f} & -\frac{x_i y_i}{f} & -x_i \end{bmatrix} \quad (3.3)$$

where Z_i is the distance between a camera and each point, that is often unavailable from monocular camera measures, and f means focal length. So, in many cases, this distance Z_i is often approximated as a suitable constant value Z_0 , which means the reference distance in reference frames [13].

3.3 Reference Extraction from Video Frames

In a video tracking task, it is very important to decide how to describe the reference path as some parameters. Although a set of feature points coordinates is a manageable geometric feature, there are measurement problem and matching problem. For example, we will suffer from feature mismatching and losing caused by occlusion or large move.

In our problem setting, camera move is restricted to 4-DOF, containing three degree of translational move and rotational move around depth axis, image transformation parameters including image translation, rotation and scaling are used as a video reference. Under this condition, a technique based on the frequency domain of the image sometime called as phase correlation is effective. A practical displacement estimation algorithm is shown in this chapter.

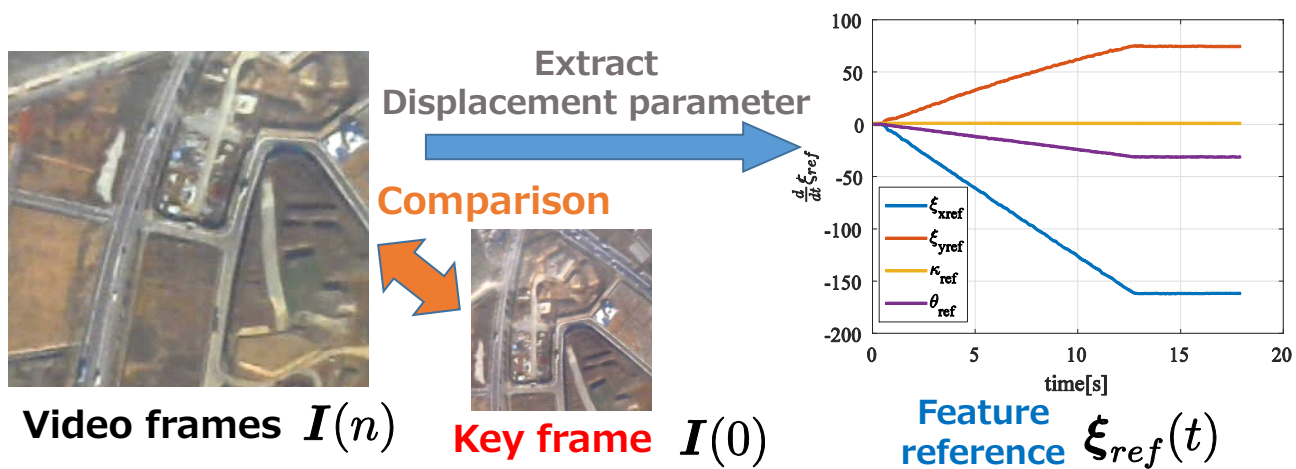


Figure 3.1 The process to extract image feature reference $\xi_{ref}(t)$ from video frames and feature reference got from this process

3.3.1 Reference Trajectory Description Based on Key Frame

Then, we use the image feature extracted with chapter 2 $\xi = (\xi_x, \xi_y, \kappa, \theta)$. Proposed method uses image displacement parameters between a particular reference image and current video frames.

More specifically, when n_{th} frame of video is defined as $I(n)$, the reference feature at n_{th} frame $\xi_{ref}(t)$ is described as the image displacement parameter from a specific frame called key frame $I(0)$. In proposed method, the first frame of the video $I(1)$ is chosen as a key frame. Fig. 3.1 shows the reference extraction flow.

3.3.2 Reference Concussion Using Relay Images

When robot moves further from initial pose, it often happens that current frame cannot be directly compared with the key frame. In that case, simple reference concussion theory using relay images is used.

The reference feature at n_{th} frame can be calculated from reference at a relay image and a relationship between the relay image and n_{th} frame. Therefore, by choosing adequate relay images, all frames in video can be transferred to image feature reference ξ . In proposed approach, the peak of POC function shown in Fig. 2.2(a) is used to switch relay images.

3.4 Video Tracking Control Using Image Based Visual Servo Scheme

Video tracking control scheme is proposed in this section. With a feature reference ξ_{ref} extracted in chapter 3.3.1, we can apply image based visual servo control scheme shown in Eq. (3.1).

In this section, feedforward approach using constant image jacobian are proposed and its effectiveness is shown in a simulation.

3.4.1 Video Tracking with Feedforward Method

Image based control scheme shown in Eq. (3.1) is a feedback approach and it causes some delay in a video tracking. So, the feedforward approach shown in Fig. 3.2 is proposed to improve tracking performance.

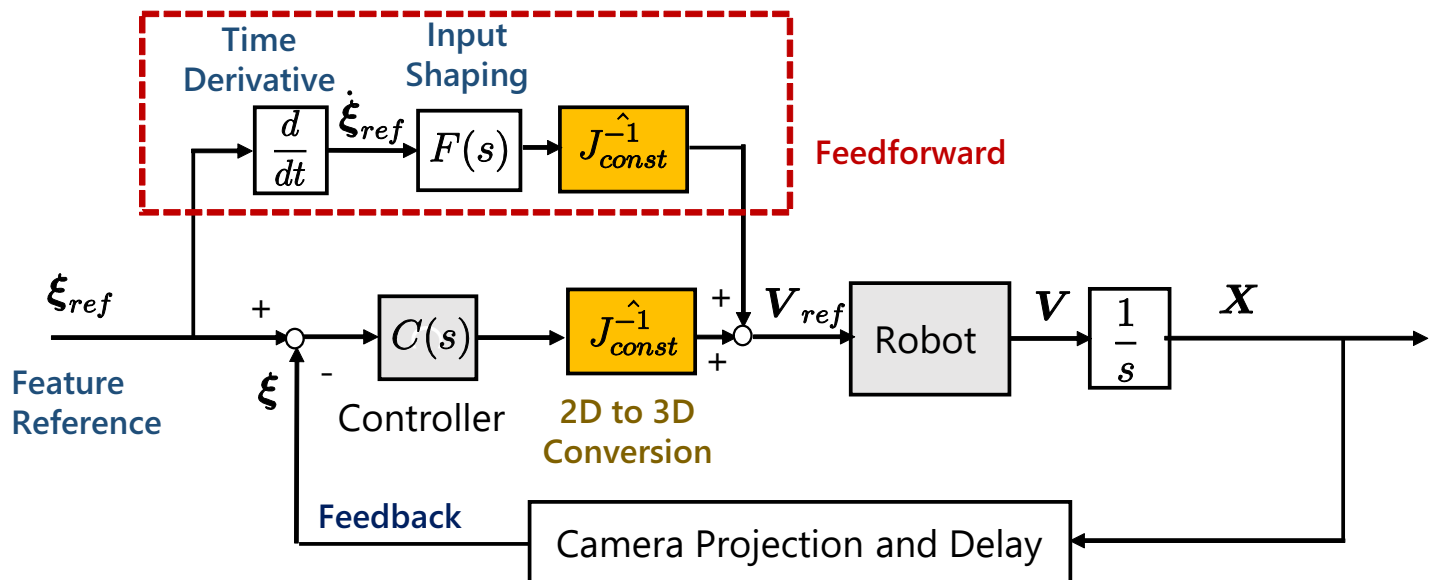


Figure 3.2 Block diagram of the proposed method.

The video feature references ξ_{ref} is calculated in advance to make a robot velocity reference V_{ref} and finally integrated to robot pose X .

In this paper, we assume that the robot response is quicker enough than sensing delay including camera imaging and image processing.

Therefore, proposed control scheme utilizing video information for feedforward is written as Eq. (3.4).

$$V_{ref}(t) = -\lambda J^+ (\xi_{ref}(t) - \xi_c(t)) + J^+ \frac{d}{dt} \xi_{ref}(t) \quad (3.4)$$

$J^+ \frac{d}{dt} \xi_{ref}(t)$ is feedforward term that compensate the change of feature reference, which needs accuracy for making proper move.

Generally, J is time-variant and difficult to estimate like shown in Eq. (3.3), our approach solve this problem by using time-invariant J_{const} in Eq. (3.10) instead. Therefore, Eq. (3.4) can be revised as Eq. (3.5):

$$V_{ref}(t) = -\lambda J_{const}^{-1} (\xi_{ref}(t) - \xi_c(t)) + J_{const}^{-1} \frac{d}{dt} \xi_{ref}(t) \quad (3.5)$$

3.4.2 Derivation of Time-Invariant Image Jacobian [1]

Surprisingly, J can be time-invariant matrix with a certain coordinate transformation in our situation: 4 DOF and planer objects.

This section shows that image jacobian matrix can be described as time-invariant matrix in a particular case that the camera move has only 3D translational and rotation around depth axis. In this section, the relationships between relative camera pose from object $X = (X, Y, Z, \Theta)$ and image transformation parameters $\xi = (\xi_x, \xi_y, \kappa, \theta)$ are derived following the image based method. X and Y means parallel axis to the planer object and Z is vertical and depth axis.

A relative camera pose in the key frame used in chapter 3.3.1 is defined as $X_0 = (X_0, Y_0, Z_0, \Theta_0)$.

Rotation and scaling displacements from the key frame are independent from each other, and correspond

Table 3.1 Variables in a simulation.

Initial Pose $(X_1, Y_1, Z_1, \Theta_1)$	$(0.2, -0.2, 0.5, 1)$ [m,rad]
Position reference $\xi_{ref}(t)$	$(R \cos \omega t, R \sin \omega t, Rt/5, 0)$ [m,rad]
Variable for spiral moves (R, ω)	$(0.25, 2 * \pi/5)$ [m,rad]
Feedback gain λ	0
Sampling Time	1 [ms]

to relative camera rotation and depth as following equations:

$$\theta = \Theta_0 - \Theta \quad (3.6)$$

$$\kappa = \frac{Z_0}{Z} \quad (3.7)$$

The point in this method is that to describe translational displacements on the coordinate systems in the reference frame. The relationship between relative camera pose and 2D translation detected in chapter 3.3.1 can be written as below:

$$\begin{pmatrix} \xi_x \\ \xi_y \end{pmatrix} = \frac{Z_0}{f} \mathbf{R}(-\Theta_0) \begin{pmatrix} X - X_0 \\ Y - Y_0 \end{pmatrix} \quad (3.8)$$

where, $\mathbf{R}(\phi)$ is a 2×2 rotation matrix representing for ϕ rotate, and f is focal length.

An image jacobian matrix can be derived from time derivative equations of Eq. (3.6), Eq. (3.7) and Eq. (3.8). Using inverse of scaling $1/\kappa$ as a feature, time derivative of Eq. (3.7) can be written in more simpler way. Finally, we choose a image feature ξ as $\xi = (\xi_x, \xi_y, 1/\kappa, \theta)$ and then we have equation in Eq. (3.9).

$$\begin{bmatrix} \dot{\xi}_x \\ \dot{\xi}_y \\ \dot{\left(\frac{1}{\kappa}\right)} \\ \dot{\theta} \end{bmatrix} = \mathbf{J}_{const} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Z} \\ \dot{\Theta} \end{bmatrix} \quad (3.9)$$

In Eq. (3.9), the image jacobian matrix can be written as time-invariant matrix expressed as follows:

$$\mathbf{J}_{const} = \begin{bmatrix} \frac{f}{Z_0} \cos \Theta_0 & \frac{f}{Z_0} \sin \Theta_0 & 0 & 0 \\ -\frac{f}{Z_0} \sin \Theta_0 & \frac{f}{Z_0} \cos \Theta_0 & 0 & 0 \\ 0 & 0 & \frac{1}{Z_0} & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad (3.10)$$

here, Θ_0 and Z_0 are relative camera pose parameters at the key frame. Θ_0 can be easily detected from Eq. (3.6), while Z_0 need some estimation methods.

3.4.3 Comparison in Simulation

Then, we compare the proposed method with conventional method shown in Eq. (3.4) using time-variant image jacobian. In this simulation, reference camera move is defined as spiral motion that includes circular

movement in X, Y axis and uniform linear motion in Z axis.

Simulation variables are shown in Table. 3.1. Against the proposed method in Eq. (3.5), feature points based method using Eq. (3.4) are used.

According to Eq. (3.3), the image jacobian \mathbf{J} for 4-DOF move used in this simulation is written as Eq. (3.11).

$$\mathbf{J} \simeq \begin{bmatrix} \frac{-1}{Z_0} & 0 & \frac{x_1}{Z_0} & y_1 \\ 0 & \frac{-1}{Z_0} & \frac{y_1}{Z_0} & -x_1 \\ \frac{-1}{Z_0} & 0 & \frac{x_2}{Z_0} & y_2 \\ 0 & \frac{-1}{Z_0} & \frac{y_2}{Z_0} & -x_2 \end{bmatrix} \quad (3.11)$$

Time-variant depth of objects from camera Z_1, Z_2 is approximated by reference depth Z_0 .

In this simulation, we suppose that Z_0 is properly estimated and set feedback gain $\lambda = 0$ to compare the accuracy of feedforward.

Fig. 3.3 and Fig. 3.4 show that the approximated image jacobian cannot make proper move, though feedback term can suppress this error.

Thus, this simulation shows proposed feedforward method using constant jacobian is better than conventional feature point based method.

3.5 Distance Estimation for Image Jacobian Estimation

3.5.1 Method 1: Scaling Parameter Based

Inverse of scaling parameter $1/\kappa$ corresponds with depth Z like shown in Eq. (3.12).

$$Z_0 = \frac{\Delta Z}{\Delta 1/\kappa} \quad (3.12)$$

here, Δ means tiny variation and this equation shows that Z_0 can be estimated from Z axis move and scaling variation. Recursive least-squares can be applied to solve this estimation. There is a problem that scaling parameter κ is often around 1.0 and has small variation, so this estimation is often sensitive to image noise.

3.5.2 Method 2: Translation Parameter Based

There is the other method using the relationship between image translation and camera translation move. Image translation and 3D translation are related to reference depth and focal length.

$$Z_0 = f \frac{\sqrt{\Delta X^2 + \Delta Y^2}}{\sqrt{\Delta \xi_x^2 + \Delta \xi_y^2}} \quad (3.13)$$

Recursive least-squares can be also applied to solve this estimation. To solve this equation, the priori information about focal length is needed.

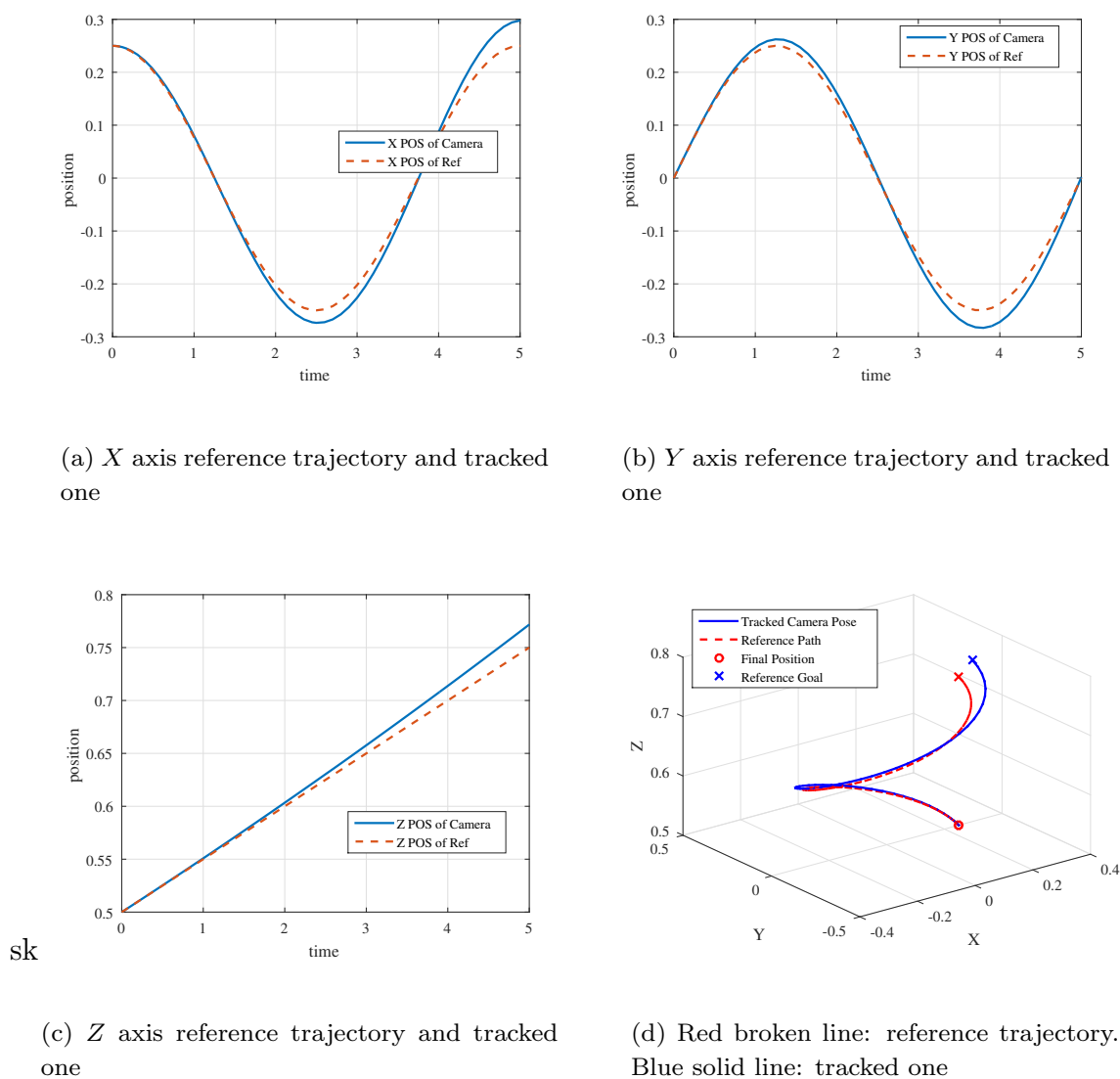


Figure 3.3 Tracking result using approximated jacobian in Eq. (3.4)

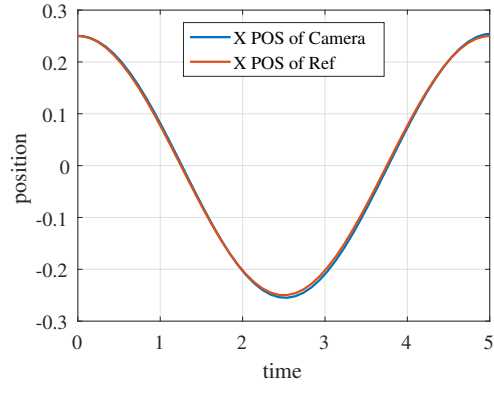
3.5.3 Distance Estimation Experiment

Using experimental data, distance is estimated with practically measured data using chapter 3.5.1 and 3.5.2. Both estimation methods using RLS program and forgetting factor is set to 0.95 at first and then converged to 1.

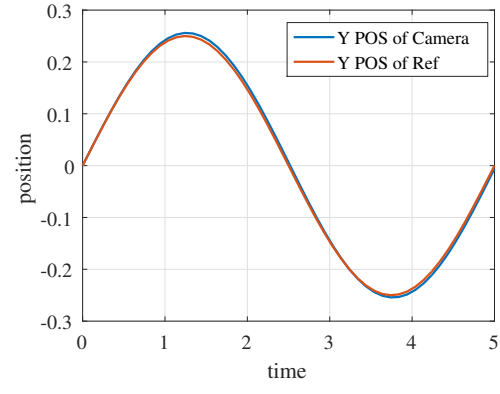
As Fig. 3.5 shows, scaling based method become oscillatory because it is sensitive to scaling changes. On the other hand, translation based method is much more stable but depends on focal length f that is a camera model parameter. So, there is a need of further approach using advanced filtering algorithms such as kalman filter.

3.6 Experiment

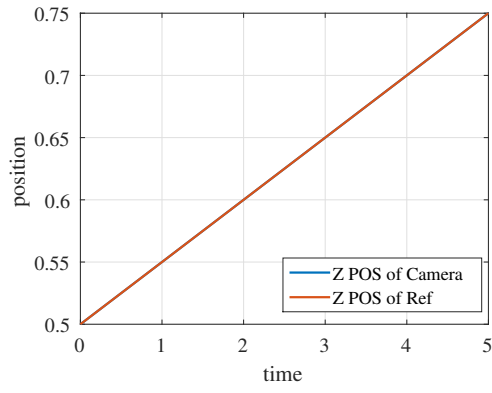
We conducted verification experiments on the contents proposed in each chapter. A part of the experimental setup is shown in Fig. 3.6. Using 6-DOF robot manipulator are driven by RTLinux, then camera is mounted on the tip of robot and a laptop computer do an image processing and a socket communication with RTLinux PC. Large part of image processing are made of C++.



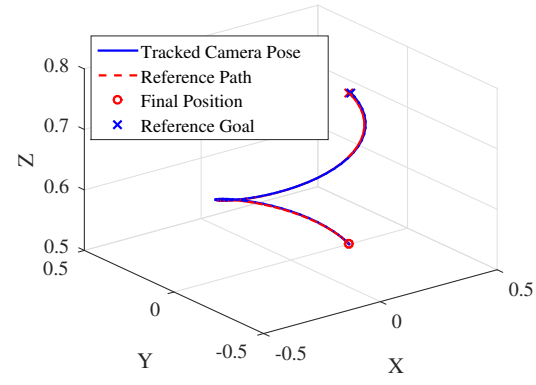
(a) X axis reference trajectory and tracked one



(b) Y axis reference trajectory and tracked one



(c) Z axis reference trajectory and tracked one



(d) Red broken line: reference trajectory. Blue solid line: tracked one

Figure 3.4 Tracking result using proposed time-invariant jacobian in Eq. (3.5)

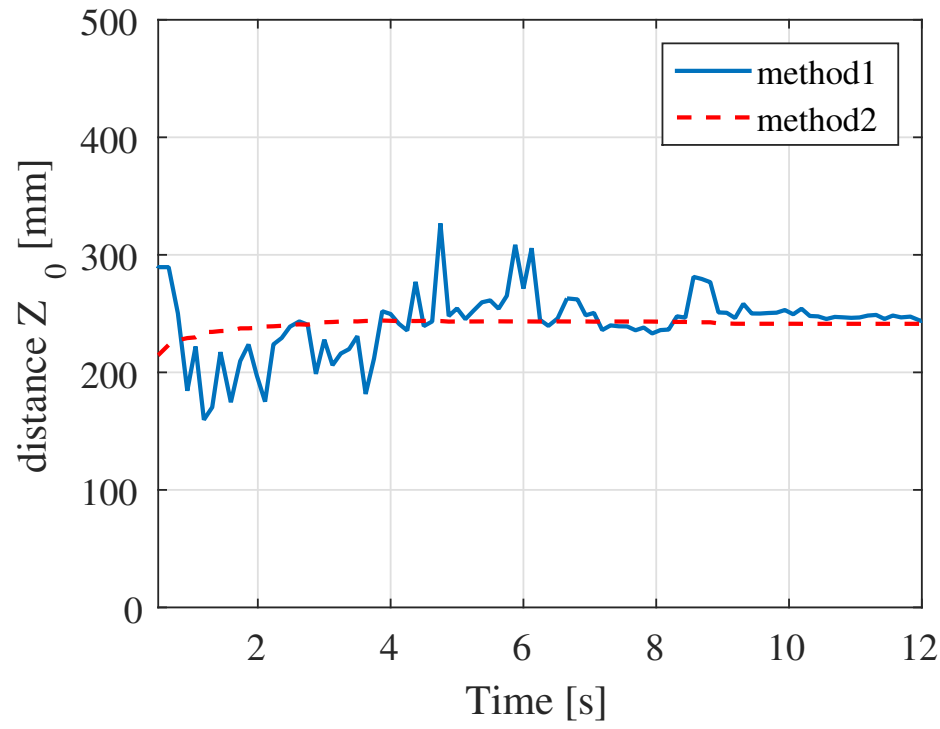


Figure 3.5 Experimental results of distance estimation. Blue line: method using scaling, red line: method using translation

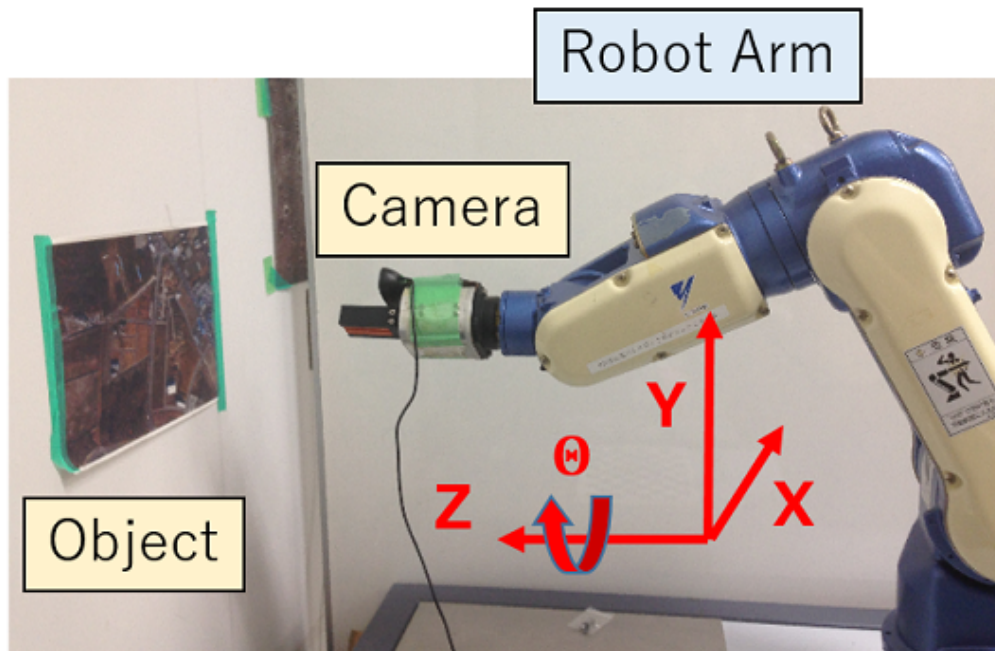


Figure 3.6 Experimental setup

For saving computational time, only 256 times 256 pixels of image are used as a control input.

Reference video is taken with the camera on robot shown in Fig. 3.6, and reference move is constant velocity move.

3.6.1 Reference Shaping and Filtering

Another problem in reference shaping and filtering is found in practical reference tracking. The problem is that the time variance of reference $\frac{d}{dt}\xi_{ref}(t)$ often becomes noisy like Fig. 3.7.

This noise causes oscillatory motion of the robot, that often hinders image acquisition and finally results in a false tracking. At this time, a 10 point moving average filter is applied to the $\frac{d}{dt}\xi_{ref}(t)$ in Fig. 3.7. The smoothed reference in Fig. 3.8 is used instead.

There should be some reasonable filtering decision frameworks for more precise tracking.

3.6.2 Experiment and Evaluation

To evaluate the proposed control law in Eq. (3.5), feedback only method in Eq. (3.1) are chosen as a comparison method.

The feedback gain λ in Eq. (3.1) and Eq. (3.5) is manually tuned to $\lambda = 1.5$. Considering frame rate of camera, image processing time and communication time, sampling period of camera is set to 125ms.

Fig. 3.9 shows the results with the comparison method. Blacked lines mean reference and solid lines are actual image feature trajectories. It is obvious that tracking error cannot be suppressed without feedforward control.

On the other hand, Fig. 3.10 shows that the proposed control scheme can compensate tracking error compared with Fig. 3.9. However, it still have some overshoot in Fig. 3.10 and move is still noisy.

There should be some improvement in filtering and reference extraction part and sensing part including image feature extraction.

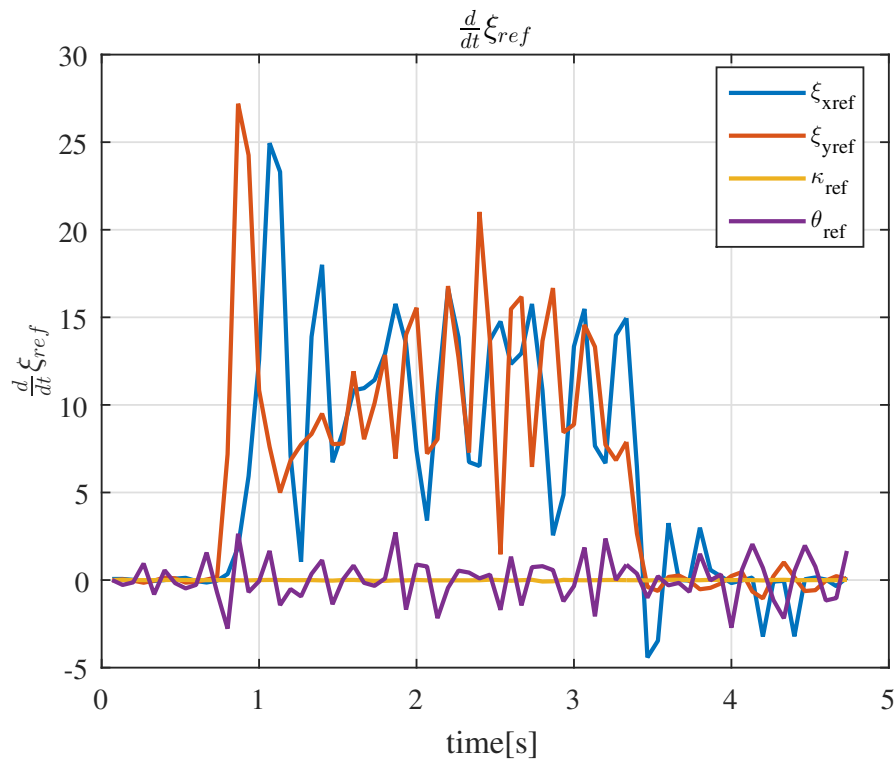


Figure 3.7 Raw time derivative of renference $\frac{d}{dt}\xi_{ref}(t)$ in Fig. 3.1.

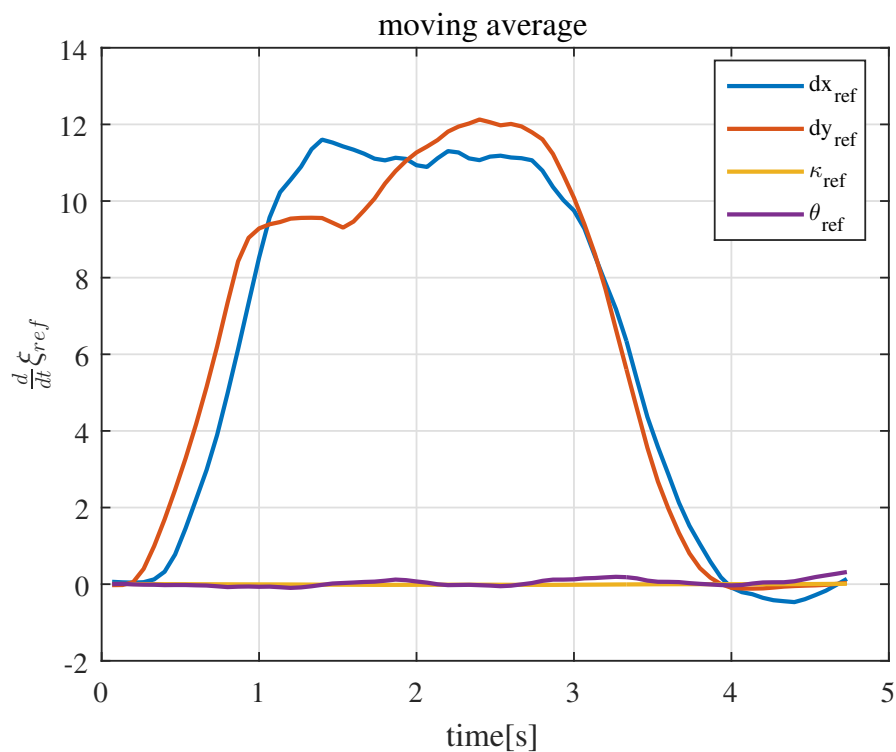


Figure 3.8 Smoothed reference $\frac{d}{dt}\xi_{ref}(t)$ by filtering Fig. 3.7

3.7 Conclusion

In this paper, the method for video tracking tasks with monocular camera was proposed. There are two main issues in this task: the process for making reference and the control scheme for reference tracking. Since these two processes are closely related each other, an integrated method design is necessary.

Proposed method is mainly aiming to utilize time-invariant image jacobian for image-based feedfor-

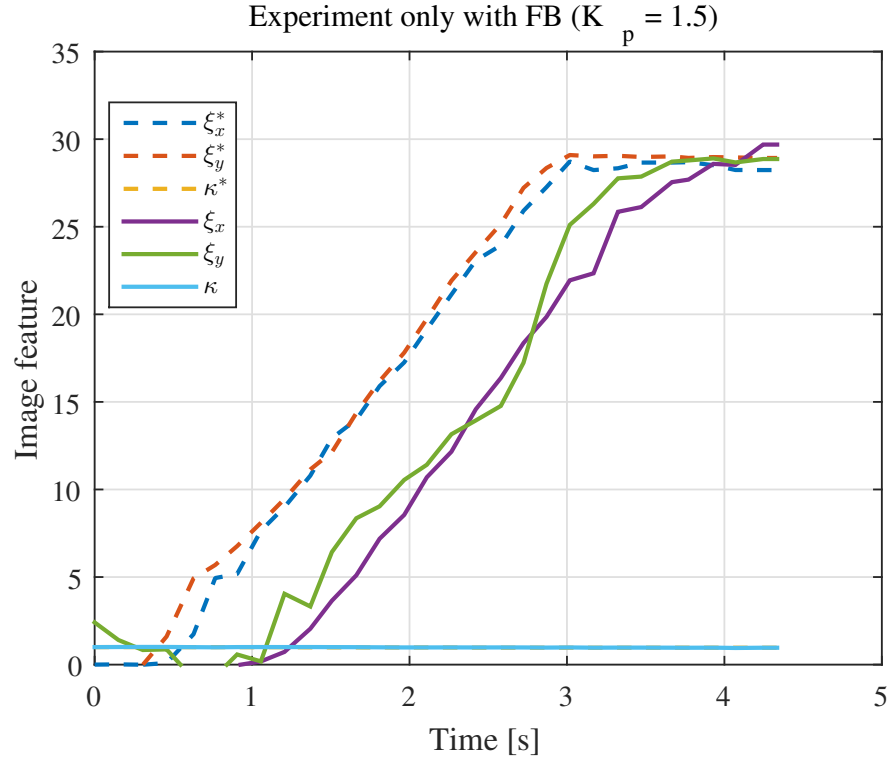


Figure 3.9 Experimental results without feedforward control. Image feature reference: broken line. Actual feature trajectories: solid line.

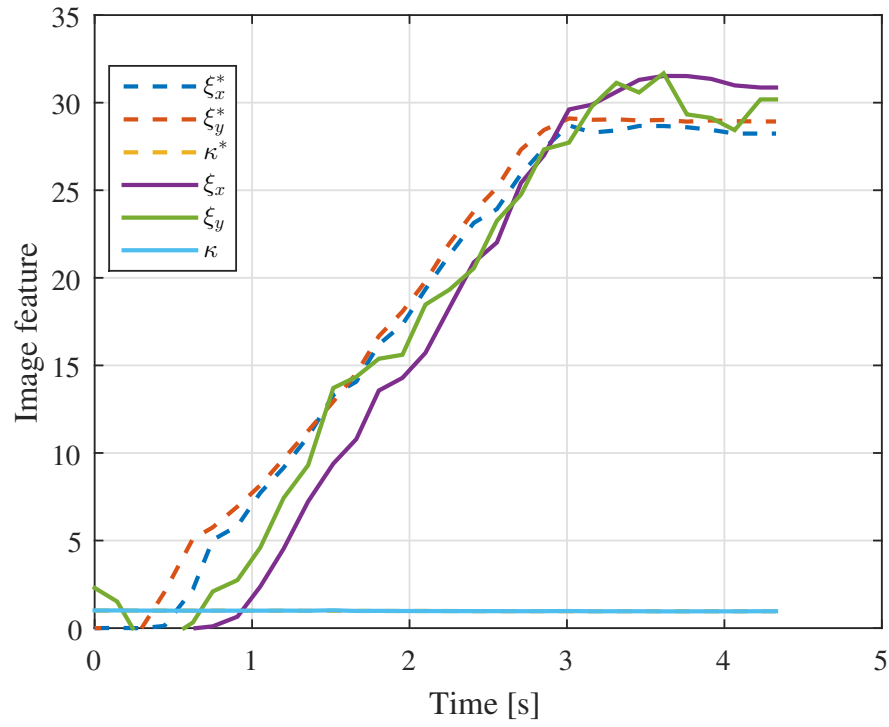


Figure 3.10 Experimental results with proposed feedforward control. Image feature reference: broken line. Actual feature trajectories: solid line.

ward control. In the image based video tracking approach, feedforward control is necessary to catch up the reference move and accurate estimation of image jacobian is needed. The time-invariant image jacobian matrix derived in this paper can solve those problem more easily and simple experiment shows its effectiveness.

Chapter 4

Drift-free Motion Estimation from Video Images using Phase Correlation and Linear Optimization

4.1 Motivation

Along with progress in device processing and computer vision technologies, a vision sensor comes to play a more important role in an environment recognition for robots. Especially, one of the popular usages of the camera is to record the robot self-motion, and this video information can be used to teach another robot how to move. Traditional image feedback control, so-called visual servo [13], is suitable for achieving video based path tracking [37] mere from a single video. The authors have revealed that feedforward with some preknowledge is essential to achieve real-time reference tracking [39], that allows a robot to copy the motion from video [40].

Since the time-series positions of the camera from video images are used in this motion tracking, it is a very important to extract the accurate position of a camera from video images. In this paper, we assume a target object as a planer and restrict 2D image motion from the camera motion as 4 degrees of freedom: 2D translation, rotation, and scaling. This assumption represents situations such as aerial images observation from UAV [41] or circuit board processing and manipulation. The easiest way to estimate camera pose is to integrate a difference from the previous frame, but it is obvious this method will have a drift error.

Then our objective is to estimate optimal motion in each frame from the whole relationship among other frames. It is almost the same idea with so-called bundle adjustment [42], which estimates the camera pose from multiple scene observations.

The problems in the bundle adjustment lie in that it uses feature points which often suffer from mismatches and non-linear equation such as the projective transformation. We use transformation parameters estimated from phase-only correlation (POC) [5] and they enable us to extract transformation parameters and its reliability without extracting any local feature. And by doing some modifications to these parameters, our optimization can be solved by a linear equation. Furthermore, the interesting linear least square solution to save computer memory and calculation time using distance matrix is revealed.

4.2 Optimization of Translational, Rotational and Scaling

Our objective is to estimate each frame's transformation parameters $\xi_{1i} = (\xi_{x_{1i}}, \xi_{y_{1i}}, \theta_{1i}, \kappa_{1i})^\top$ from the origin. We set the initial video frame as a origin $\xi_{11} = (0, 0, 0, 1)^\top$ and call it a key frame.

Then, we have measurements of whole relationships between possible two frames $\hat{\xi}_{ij}$ ($i < j$), so the cost function to be minimized looks like Eq. (4.1).

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^n \omega_{ij} \|\xi_{ij} - \hat{\xi}_{ij}\|^2 \quad (4.1)$$

In Eq. (4.1), ξ_{ij} is a function of ξ_{1i} and ξ_{1j} defined in Eq. (3.8), and ω_{ij} means a weight for this measurement.

Although it is possible to use non-linear optimization method to minimize Eq. (4.1), we can use a linear least square method by optimizing each translation, rotation and scaling separately. This is possible because rotation and scaling transformations estimated with POC are independent of other parameters.

4.2.1 Rotation Optimization.

First, we begin with rotation optimization. When i^{th} frame of video image has θ_{1i} rotational transformation from the key frame and j^{th} one has θ_{1j} , measured rotational transformation between those images θ_{ij} can be written as those difference:

$$\theta_{ij} = \theta_{1j} - \theta_{1i} \quad (4.2)$$

Therefore, substituting θ_{ij} to the ξ_{ij} in Eq. (4.1), the cost function becomes a least square equation and easily can be solved.

4.2.2 Scaling Optimization.

Subsequently, scaling optimization is performed. When i^{th} frame and j^{th} frame have scaling transformation κ_{1i} and κ_{1j} from the key frame, measured transformation κ_{ij} can be expressed with division:

$$\kappa_{ij} = \frac{\kappa_{1j}}{\kappa_{1i}} \quad (4.3)$$

By taking logarithm of Eq. (4.3), this equation can be converted to Eq. (4.4).

$$\log \kappa_{ij} = \log \kappa_{1j} - \log \kappa_{1i} \quad (4.4)$$

So, considering $\mu_{ij} = \log \kappa_{ij}$ enables us to use the same method with chapter 4.2.1. Then estimated scaling can be reconstructed from the equation $\hat{\kappa}_{1i} = \exp(\hat{\mu}_{1i})$.

4.2.3 Translation Optimization

Translation optimization is done after rotation and scaling optimization. A measured translation from i^{th} frame to j^{th} frame $(\xi_{x_{ij}}, \xi_{y_{ij}})$ has relationships with the rotation θ_{1i} and the scaling κ_{1i} calculated

in previous steps. This is because the translation is defined in reference frame so each translation is described on different coordinates.

Then, the relationships among those parameters can be written as follows:

$$\begin{pmatrix} \xi_{x_{ij}} \\ \xi_{y_{ij}} \end{pmatrix} = \frac{1}{\kappa_{1i}} \begin{pmatrix} \cos \theta_{1i} & -\sin \theta_{1i} \\ \sin \theta_{1i} & \cos \theta_{1i} \end{pmatrix} \begin{pmatrix} \xi_{x_{1j}} \\ \xi_{y_{1j}} \end{pmatrix} - \begin{pmatrix} \xi_{x_{1i}} \\ \xi_{y_{1i}} \end{pmatrix} \quad (4.5)$$

This equation allows vertical and horizontal translation interfering each other, thus variable conversion in Eq. (4.6) is applied to eliminate this interference.

$$\begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix} = \kappa_{1i} \begin{pmatrix} \cos \theta_{1i} & \sin \theta_{1i} \\ -\sin \theta_{1i} & \cos \theta_{1i} \end{pmatrix} \begin{pmatrix} \xi_{x_{ij}} \\ \xi_{y_{ij}} \end{pmatrix} \quad (4.6)$$

Note that this transformation become an identity transformation in $i = 1$ because of $\theta_{11} = 0$ and $\kappa_{11} = 1$, so $(\xi_{x_{1i}}, \xi_{y_{1i}}) = (x_{1i}, y_{1i})$. By applying a transformation in Eq. (4.6), Eq. (4.5) can be rewritten as follows:

$$\begin{pmatrix} x_{ij} \\ y_{ij} \end{pmatrix} = \begin{pmatrix} x_{1j} \\ y_{1j} \end{pmatrix} - \begin{pmatrix} x_{1i} \\ y_{1i} \end{pmatrix} \quad (4.7)$$

Eq. (4.7) shows that both x and y is independent and these parameters can be optimized using Eq. (4.14).

Thus, all of 4 estimated parameters can be optimized with a linear least square method including simple trick shown in this section.

4.3 Solving Least Square Problem via Distance Matrix

By discussion in chapter 4.2, all we have to do is just solve the linear least square problem and its equation is represented as Eq. (4.10). The problem is that the number of relationships among video frames increases in the second order of total frame number n . This section shows the new method to use a distance matrix to save a calculation cost.

4.3.1 Definition of Distance Matrix

A distance matrix is often used in graph theory, containing distance information between the elements of some data set. In the image registration problem, distance matrix has been used to estimate mean images from test data [43].

In this paper, we consider points cloud on a one-dimensional line. As shown in upper side of Fig. 4.1, coordinates of points are defined as p_i ($i = 1, 2, 3, \dots, n$) and a distance between p_i and p_j as d_{ij} ($j > i$). These two parameters have relationships like Eq. (4.8).

$$d_{ij} = p_j - p_i \quad (j \geq i) \quad (4.8)$$

Note p_1 is a reference point that is needed to define other points' coordinates and its coordinate is 0.

Then, our distance matrix in this model \mathbf{D} can be written as Eq. (4.9).

$$D(i, j) = \begin{cases} d_{ij} & (j > i) \\ 0 & (otherwise) \end{cases} \quad (4.9)$$

$D(i, j)$ means a component in i^{th} row and j^{th} column of \mathbf{D} and stores a distance information between p_i and p_j . With this distance matrix form, we can graphically see relationships between those points; for example, any distance from p_i is represented in i^{th} row and column of \mathbf{D} .

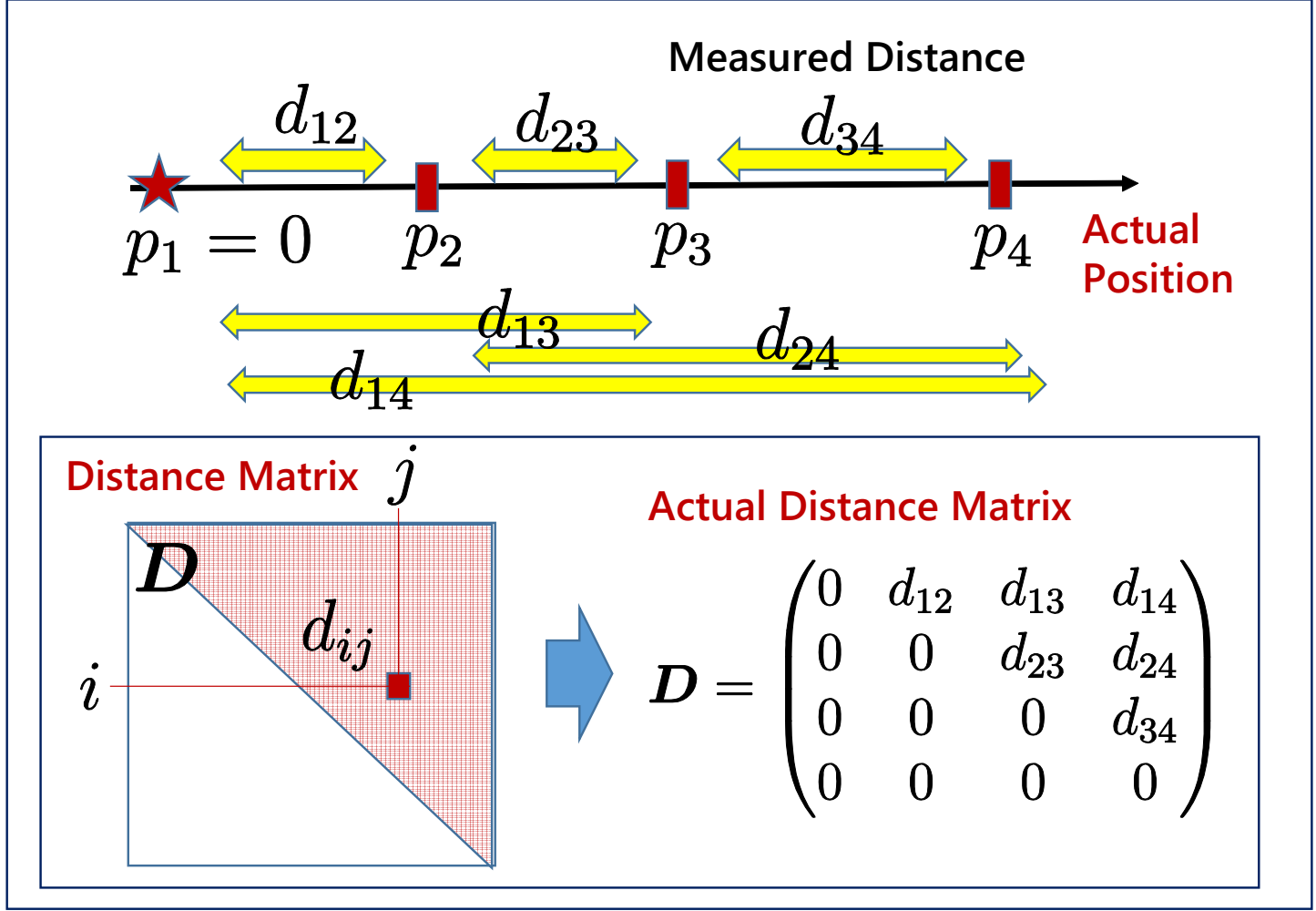


Figure 4.1 An example of a distance matrix D shown in Eq. (4.9)

Our goal is to estimate the parameters p_i ($i = 2, 3, \dots, n$) from those distances d_{ij} measured with some observation or processing.

In this section, the way to solve this least squares problem with the distance matrix D is proposed.

4.3.2 Conversion to the Least Squares Problem

Our objective can be represented as minimizing the sum of error between measured distances d_{ij} and estimated one $p_j - p_i$ from Eq. (4.8). The sum of error J can be written as follows:

$$J = \sum_{i=1}^{n-1} \sum_{j=i+1}^n \omega_{ij} ((p_j - p_i) - d_{ij})^2 \quad (4.10)$$

while, ω_{ij} means a weight for a measurement of d_{ij} . When the parameter vector is set to $\mathbf{x} = (p_2, \dots, p_n)^\top$, Eq. (4.10) can be rewritten as a following weighted linear least square problem:

$$J = (\mathbf{Ax} - \mathbf{y})^\top \mathbf{W} (\mathbf{Ax} - \mathbf{y}) \quad (4.11)$$

$$\mathbf{y} = \begin{pmatrix} d_{12} \\ \vdots \\ d_{1n} \\ d_{23} \\ \vdots \\ d_{2n} \\ \vdots \\ d_{(n-1)n} \end{pmatrix}, \mathbf{A} = \begin{pmatrix} 1 & 0 & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & 0 & 1 \\ -1 & 1 & \cdots & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & \cdots & \cdots & 0 & 1 \\ \vdots & & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & \cdots & -1 & 1 \end{pmatrix} \quad (4.12)$$

$$\mathbf{W} = \text{diag}(\omega_{12}, \dots, \omega_{1n}, \omega_{23}, \dots, \omega_{2n}, \dots, \omega_{(n-1)n}) \quad (4.13)$$

In Eq. (4.11), \mathbf{y} is a $\frac{n(n-1)}{2}$ rows measurement vector and \mathbf{A} is a $\frac{n(n-1)}{2} \times n-1$ size coefficient matrix.

Also, $\text{diag}(\boldsymbol{\eta})$ means diagonal matrix of a vector $\boldsymbol{\eta}$ and \mathbf{W} is a $\frac{n(n-1)}{2} \times \frac{n(n-1)}{2}$ weight matrix.

Then, approximated value $\hat{\mathbf{x}}$ minimizing the J in Eq. (4.11) can be expressed like Eq. (4.14).

$$\hat{\mathbf{x}} = (\mathbf{A}^\top \mathbf{W} \mathbf{A})^{-1} \mathbf{A}^\top \mathbf{W} \mathbf{y} \quad (4.14)$$

However, when the number of parameter n is increased, the size of matrix \mathbf{A} and \mathbf{W} become very large and consume very large amount of memory. For example, when frame number is $n = 351$, the size of coefficient matrix \mathbf{A} become 61425×350 .

Then, to avoid to calculate \mathbf{A} , transform Eq. (4.14) to another form called a normal equation:

$$(\mathbf{A}^\top \mathbf{W} \mathbf{A}) \hat{\mathbf{x}} = \mathbf{A}^\top \mathbf{W} \mathbf{y} \quad (4.15)$$

In Eq. (4.15), $\mathbf{A}^\top \mathbf{W} \mathbf{A}$ and $\mathbf{A}^\top \mathbf{W} \mathbf{y}$ can be directly calculated by solving conditions of the saddle point $\frac{\partial J}{\partial p_i} = 0$ ($i = 2, 3, 4, \dots, n$). The result is shown in Eq. (4.16). $\mathbf{A}^\top \mathbf{W} \mathbf{A}$ is $n-1 \times n-1$ matrix and $\mathbf{A}^\top \mathbf{W} \mathbf{y}$ is $n-1$ length vector, therefore in term of memory reduction it is more efficient way.

$$\begin{aligned}
\mathbf{A}^\top \mathbf{W} \mathbf{A} &= \begin{pmatrix} \omega_{12} + \sum_{j=3}^n \omega_{2j} & \cdots & -\omega_{2k} & \cdots & -\omega_{2n} \\ \vdots & \ddots & \vdots & & \vdots \\ -\omega_{2k} & \cdots & \sum_{i=1}^{k-1} \omega_{ik} + \sum_{j=k+1}^n \omega_{kj} & \cdots & -\omega_{kn} \\ \vdots & & \vdots & \ddots & \vdots \\ -\omega_{2n} & \cdots & -\omega_{kn} & \cdots & \sum_{i=1}^{n-1} \omega_{in} \end{pmatrix} \\
\mathbf{A}^\top \mathbf{W} \mathbf{y} &= \begin{pmatrix} \omega_{12}d_{12} - \sum_{j=3}^n \omega_{2j}d_{2j} \\ \vdots \\ \sum_{i=1}^{k-1} \omega_{ik}d_{ik} - \sum_{j=k+1}^n \omega_{kj}d_{kj} \\ \vdots \\ \sum_{i=1}^{n-1} \omega_{in}d_{in} \end{pmatrix}
\end{aligned} \tag{4.16}$$

4.3.3 Normal Equation Derivation Using Distance Matrix's Information

Then, this part shows how to calculate normal equation from distance matrix. Given we already have a distance matrix of measurement \mathbf{R} and weight matrix $\mathbf{\Omega}$ in the same form like below:

$$\begin{aligned}
\mathbf{R}(i, j) &= \begin{cases} d_{ij} & (j > i) \\ 0 & (otherwise) \end{cases} \\
\mathbf{\Omega}(i, j) &= \begin{cases} \omega_{ij} & (j > i) \\ 0 & (otherwise) \end{cases}
\end{aligned} \tag{4.17}$$

$\mathbf{A}^\top \mathbf{W} \mathbf{A}$ calculation.

the left side of Eq. (4.15) can be calculated from $\mathbf{\Omega}$ and its whole step is shown in Fig. 4.2.

$$\mathbf{A}^\top \mathbf{W} \mathbf{A} = \text{diag}(\sigma_\Omega(2:n)) - TU(\mathbf{\Omega}) - TU(\mathbf{\Omega})^\top \tag{4.18}$$

$$\sigma_\Omega = \sigma_v(\mathbf{\Omega}) + \sigma_h(\mathbf{\Omega}) \tag{4.19}$$

Each $\sigma_v(\mathbf{\Omega})$ and $\sigma_h(\mathbf{\Omega})$ respectively means vertical and horizontal sum vector respectively and $\sigma_\Omega(2:n)$ means a second to n^{th} row part of σ_Ω . In addition, $TU(\mathbf{\Omega})$ a part of $\mathbf{\Omega}$ containing 2^{nd} to n^{th} row and column; this can be written as $TU(\mathbf{\Omega}) = \mathbf{\Omega}(2:n, 2:n)$.

$\mathbf{A}^\top \mathbf{W} \mathbf{y}$ calculation.

On the other hand, $\mathbf{A}^\top \mathbf{W} \mathbf{y}$, the right hand of Eq. (4.15) can be also calculated from \mathbf{R}_w , which is element-wise multiplication of \mathbf{R} and $\mathbf{\Omega}$ and its flow is shown in Fig. 4.3. The equation can be expressed as follows:

$$\mathbf{A}^\top \mathbf{W} \mathbf{y} = \delta_{R_w}(2:n) \tag{4.20}$$

$$\delta_{R_w} = \sigma_v(R_w) - \sigma_h(R_w) \tag{4.21}$$

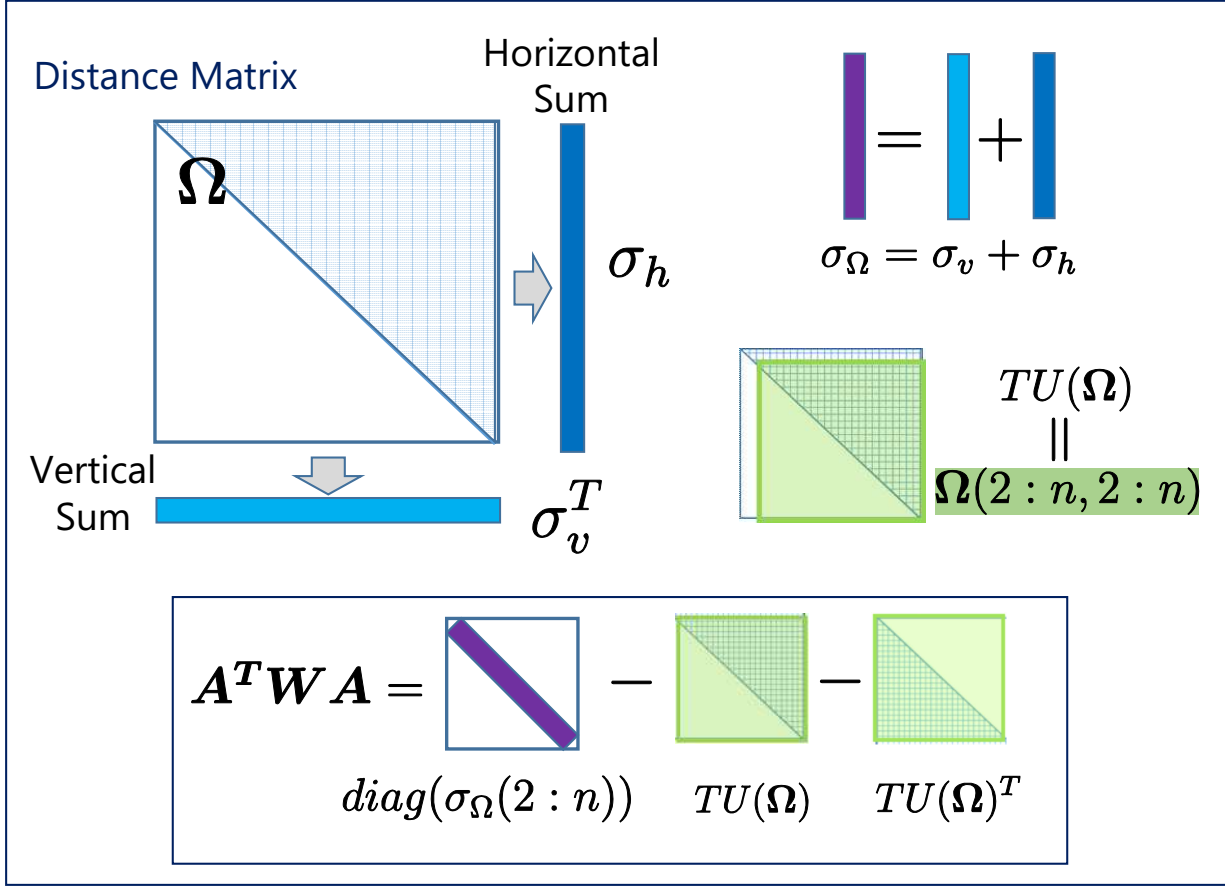


Figure 4.2 How to calculate $A^T W A$.

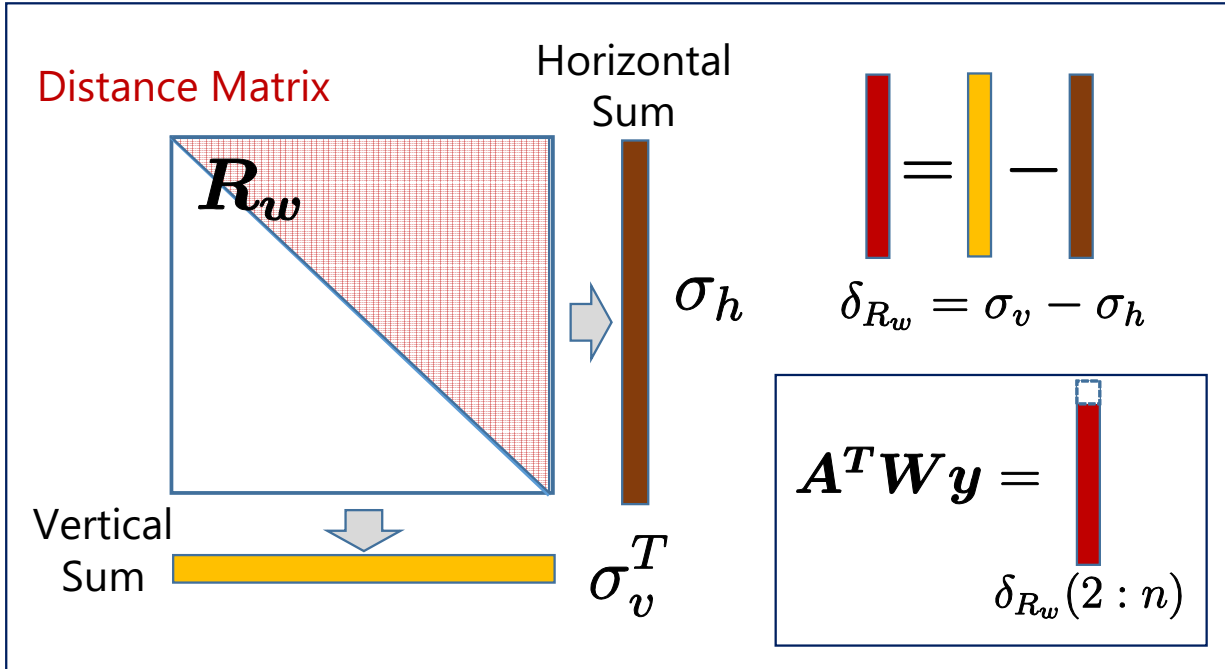
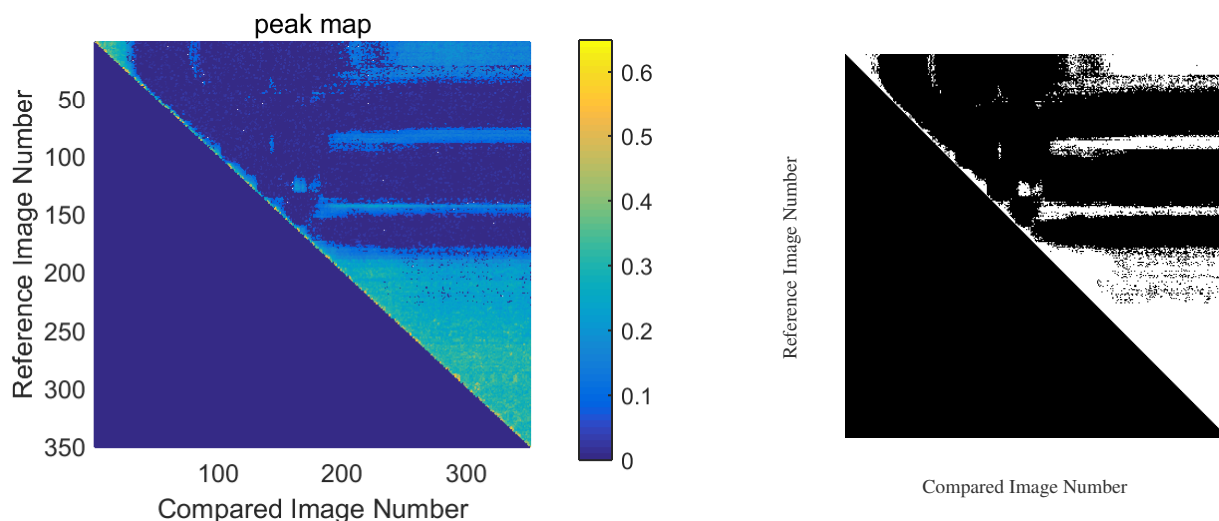


Figure 4.3 How to calculate $A^T W y$.

Thus, normal equation shown in Eq. (4.15) can be solved with distance matrix of measurement and weight matrix in Eq. (4.17).

4.4 Experiment and Evaluation

Experiments using actual video were held to confirm effectiveness of our method shown in chapter 4.2. To show the importance of this optimization, the easiest method to integrate differences is chosen as a



(a) Peak value Matrix \mathbf{P} .

(b) Distance matrix of weight $\mathbf{\Omega}$.

Figure 4.4 The way to make a weight matrix in Fig. 4.4(b) from a reliability matrix in Fig. 4.4(a).

Table 4.1 Parameters in evaluation

Processor spec	Corei7-4600U 2.1GHz
Size of each video frame	864×480 [pix]
Frame number n	351
Frame Rate	30

comparison. Evaluations were also done in both qualitative and quantitative way using image mosaicing technique. Table. 4.1 shows parameters for experimental video. Each video frame is cropped to 256×256 so that the calculation time of image processing become shorter and the quantitative evaluation is able to be achieved. Calculation is done with MATLAB 2015b on laptop windows10 PC.

Image transformation parameter between any two frames are computed in advance, and extracted relationships including POC's peak value representing matching confidence are prepared as formation of distance matrix shown in 4.3.

4.4.1 Decision of Weight $\mathbf{\Omega}$

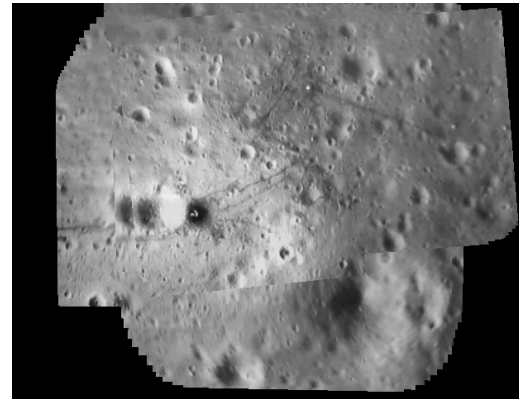
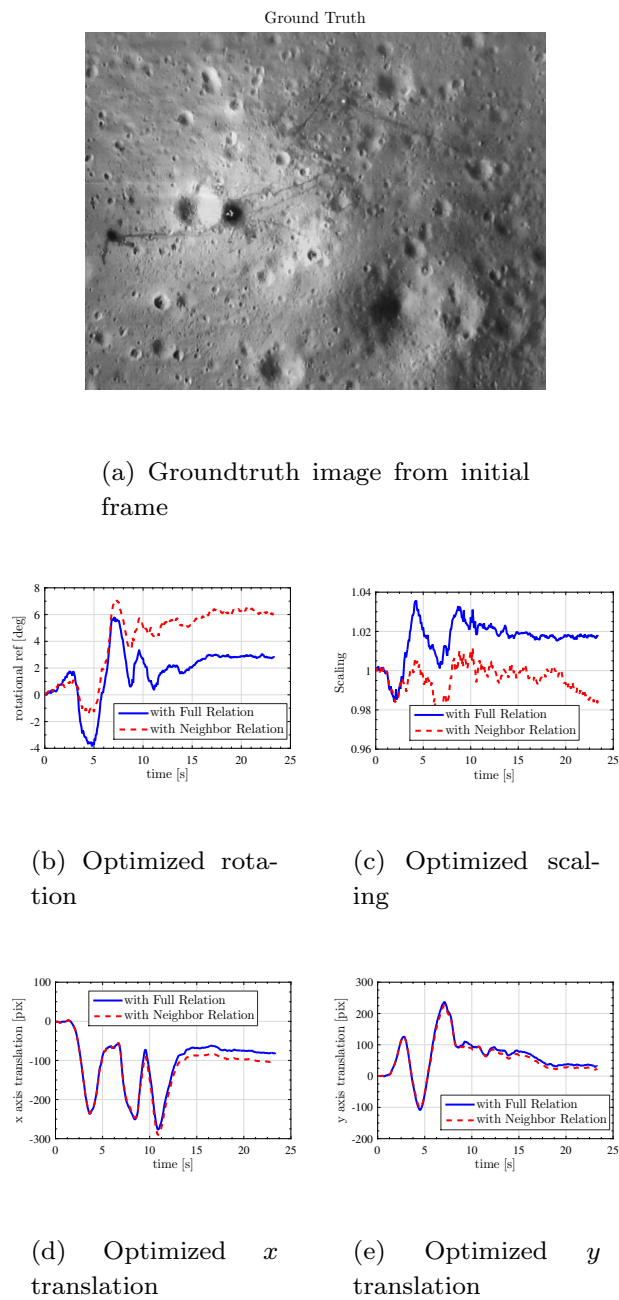
The weight of each measurement $\mathbf{\Omega}$ is decided from POC's peak value. Fig. 4.4(a) shows the matrix \mathbf{P} including POC's peak value p_{ij} between i^{th} and j^{th} frame.

The most simple way to decide weight is to use relationships which have larger peak value than threshold value p_{TH} . In this way, a binary weight matrix $\mathbf{\Omega}$ shown in Eq. (4.22) can be made from \mathbf{P} .

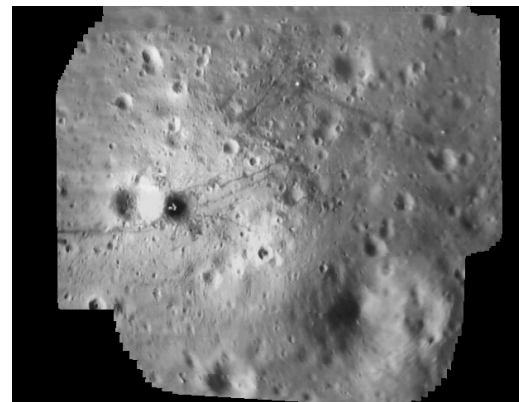
$$\omega_{ij} = \begin{cases} 1 & (p_{ij} \geq p_{Th}) \\ 0 & (p_{ij} < p_{Th}) \end{cases} \quad (4.22)$$

Then, the weight matrix $\mathbf{\Omega}$ used in this experiment is decided as Fig. 4.4(b). In this binarization, the threshold value which separates successful estimation from failure is $p_{Th} = 0.075$. This cutoff is decided empirically between 0.05 and 0.1.

From now on, the estimation method by optimization using this $\mathbf{\Omega}$ is called as proposed method, and



(f) Mosaiced image with only neighbor information



(g) Mosaiced image with the proposed method

Figure 4.5 Qualitative evaluation. (a) ground truth image used in evaluation. (b)(c)(d)(e) Optimized transformation parameter. Red broken line shows iterative method and blue line shows proposed method using distance matrix. (f) Mosaiced image with conventional method. (g) Mosaiced image with proposed method. It is obvious that the (g) is mosaiced better than (f) compared with the ground truth

the estimation method by iteratively integrating neighbor frame errors is called as conventional one. Fig. 4.5(b) to Fig. 4.5(e) show the optimized 2D camera motion with proposed method and conventional method. There are visible differences between estimated values in these two methods.

Then, the qualitative and quantitative evaluation is achieved with the image mosaicing technique.

4.4.2 Qualitative Evaluation

First, a qualitative evaluation based on image mosaicing is held. In the mosaicing process, every frame in the video is transformed based on each estimated value, and then overlapped on the image plane of 1st frame. The ground truth image for comparison is shown in Fig. 4.5(a), and it is generated from 1st

Table 4.2 Average Computation Time

With normal calculation method in Eq. (4.14)	2203[ms]
With proposed method with distance matrix	4.24[ms]

frame before cropping. The results are shown in Fig. 4.5(f) and Fig. 4.5(g). It is obvious that there is less contradiction in proposed Fig. 4.5(g) than conventional Fig. 4.5(f).

4.4.3 Quantitative Evaluation

Then, quantitative evaluation based on root mean squared error (RSE) and mutual information (MI) [37] are held. Each frame in the video is compared with the corresponding area in the ground truth. The RSE in Fig. 4.6(a) means the evaluated error in each frame from the ground truth, and the MI in Fig. 4.6(b) means similarity. In both evaluation method, the proposed method achieves smaller error and higher similarity.

4.4.4 Computation Time

We evaluate the computing time for optimization based on distance matrix method. Table. 4.2 shows the comparison with the normal linear least square solution shown in Eq. (4.14). In this comparison, the 61425×350 matrix \mathbf{A} and the weight matrix \mathbf{W} are treated as a sparse matrix so that memory and computation time can be saved. But the result shows that the proposed method is still faster.

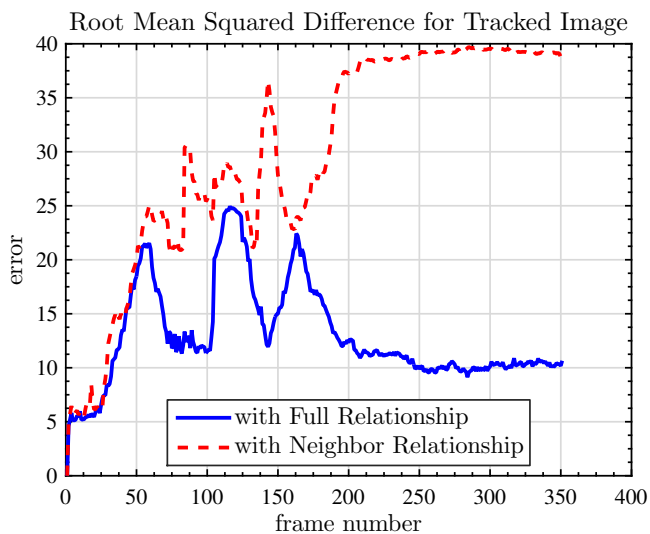
4.4.5 Comparison between feature point based method

Additionally, we tried to replace the transformation parameter estimation with SIFT and RANSAC. The difference is in the decision of weight matrix $\mathbf{\Omega}$. We used inlier rate to exclude false results and set the $\mathbf{\Omega}$ in Fig. 4.7(a). The optimization result with SIFT shown in Fig. 4.7(b) indicates some false detections can not be excluded with mere a inlier rate. Therefore we need more better method to distinguish each frame relationship is reliable. In this point, POC based measurement has a advantage on the local feature based method.

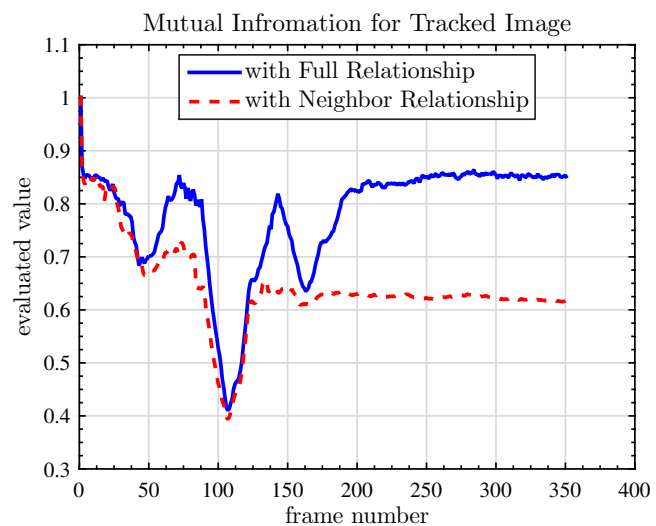
4.5 Conclusion

The main topic of this paper is to estimate the more precise camera motion by using all available relationships between video frames. Compared with the bundle adjustment approach, our proposed method does not have to solve the non-linear optimization problem. Furthermore, POC has an advantage over the feature points based method in estimating its measurement reliability. In the least square optimization, the small but interesting memory saving solution using distance matrix is proposed. It can be applied to every similar problem and save many time and computation memories. The effectiveness of proposed method is evaluated using image mosaicing technique and it is also able to apply other camera motion estimation.

The authors are going to use some dataset to compare the bundle adjustment based estimation and the proposed method in this paper. Further research will be held on an iterative method like Kalman

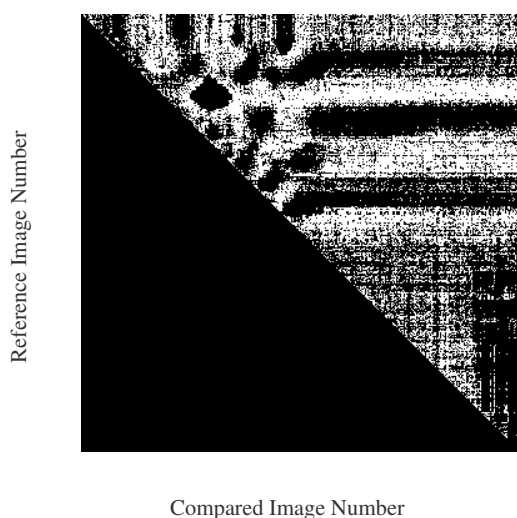


(a) RMS error for each frame

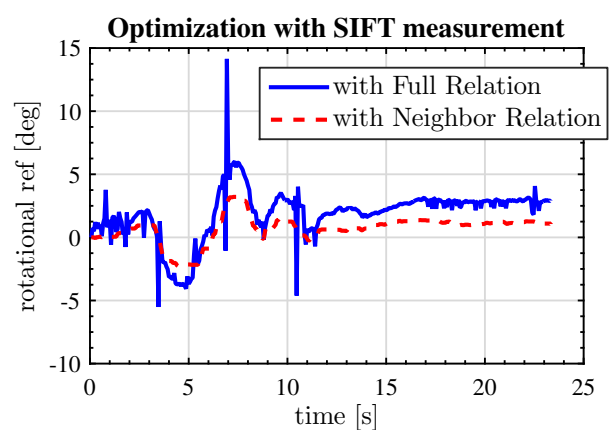


(b) Mutual Information for each frame

Figure 4.6 Quantitative evaluation. (a) RMS error between ground truth and each estimated frame. (b) Mutual Information between ground truth and estimated frame. Blue lines, which represent proposed method are better than conventional one represented as red broken lines.



(a) Weight matrix Ω with SIFT.



(b) Rotation optimization with SIFT.

Figure 4.7 The weight matrix with SIFT and optimization results from SIFT measurement.

filtering or recursive least square methods with reasonable weight function then compared with SLAM based approach [41].

Part II

Position-Based Method

Chapter 5

Ground Vehicle Position Estimation by Monocular Vision and Height Constraints

5.1 Motivation

In contrast to the image based visual servo, the position based visual servo requires precise three-dimensional relative position estimation. The 3D reconstruction from image information is a hard task since images are normally two-dimensional information and have ambiguity on the depth. Therefore, we focus on vision-based distance estimation methods.

There are mainly two types of approach to estimate depth from vision sensors:

- 1 Using prior knowledge such as hardware constraints of camera setup position or angle, or object shape and size information with some assumption.
- 2 Using another sensor information such as LIDAR or IMU. Stereo observation with multiple cameras is also included.

In this chapter, we propose hardware constraints driven position estimation method for indoor vehicle position control.

5.1.1 Setup of experimental field

As shown in Fig. 5.1, a space for small robotic vehicles was created in the indoor room. Since the GPS signal cannot be obtained indoors, the camera is placed at a sufficiently high position from the ground to measure the position. Fig. 5.1 is an image taken from the camera and cut.

Fig. 5.2 shows our solution: we consider that the position of the robot is estimated indirectly by detecting the marker placed on the back of the vehicle from the image acquired from the ceiling camera. The detailed flow of the algorithm is shown in Fig. 5.3.

5.1.2 Related works and backgrounds

Vision-based indoor positioning can be classified into two categories depending on camera positions: camera on environment or camera on the robot. The former assumes an environmental camera, such as a surveillance camera, to track markers on the robots. On the other hand, in the latter system robots use camera to observe markers on the environment.

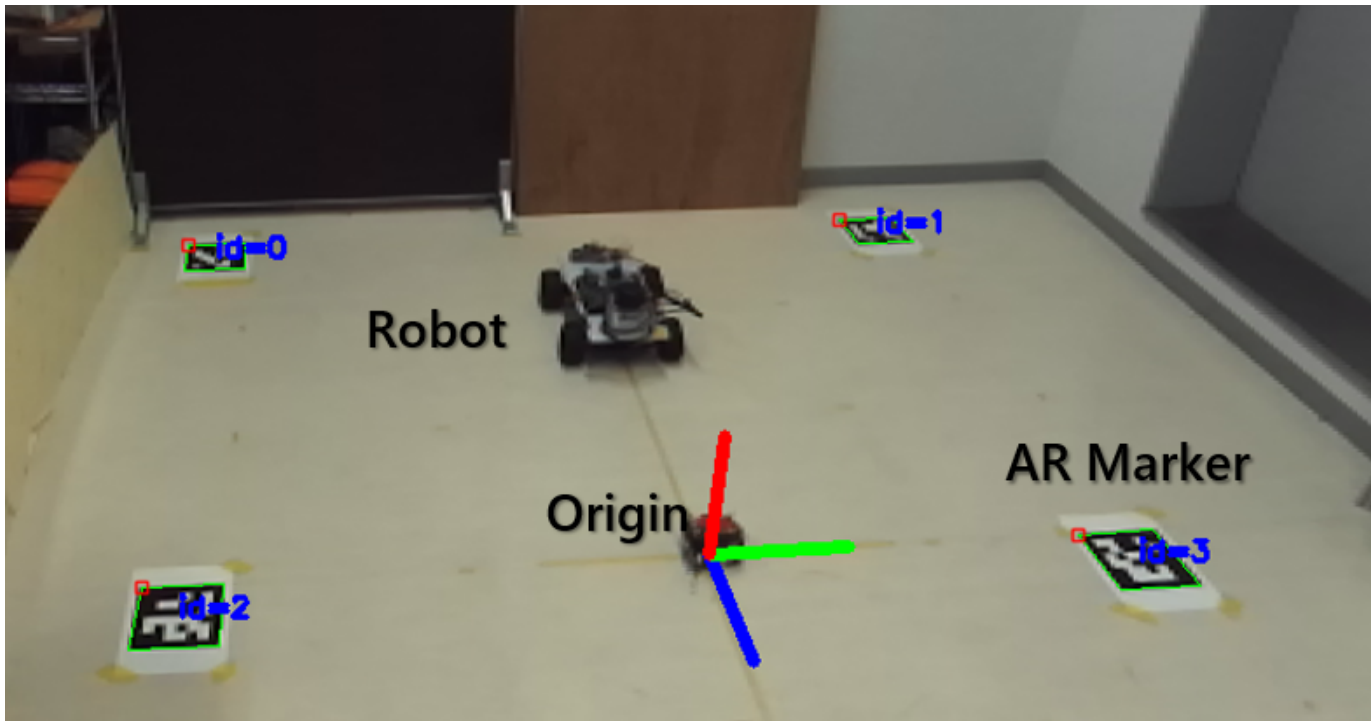


Figure 5.1 Experimental field captured from the ceil camera.

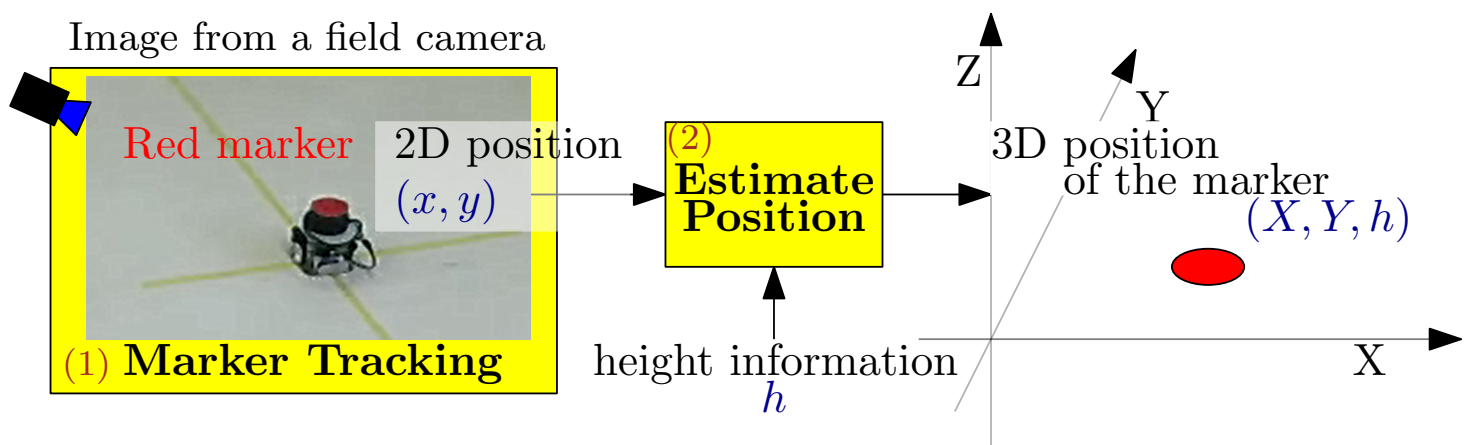


Figure 5.2 Program flow of the proposed position estimation method.

The kind of constraints and marker patterns used in these methods are different. Generally speaking, the former system often using marker size and alignments information to reconstruct the depth [44], and the latter tends to use bird-eye transformation with known camera pose to ground [45, 46].

5.2 Robust marker tracking method

At first, we started with marker tracking on the color information.

5.2.1 Marker used in this paper

When tracking the position of an indoor robot using a camera, tracking markers are often used in many research. Markers with a specific pattern and markers [47] with a specific color [48] are common.

The ArUco marker is one of the popular patterned markers with high accessibility and accuracy. We used ArUco markers to define coordinate shown in Fig. 5.1.

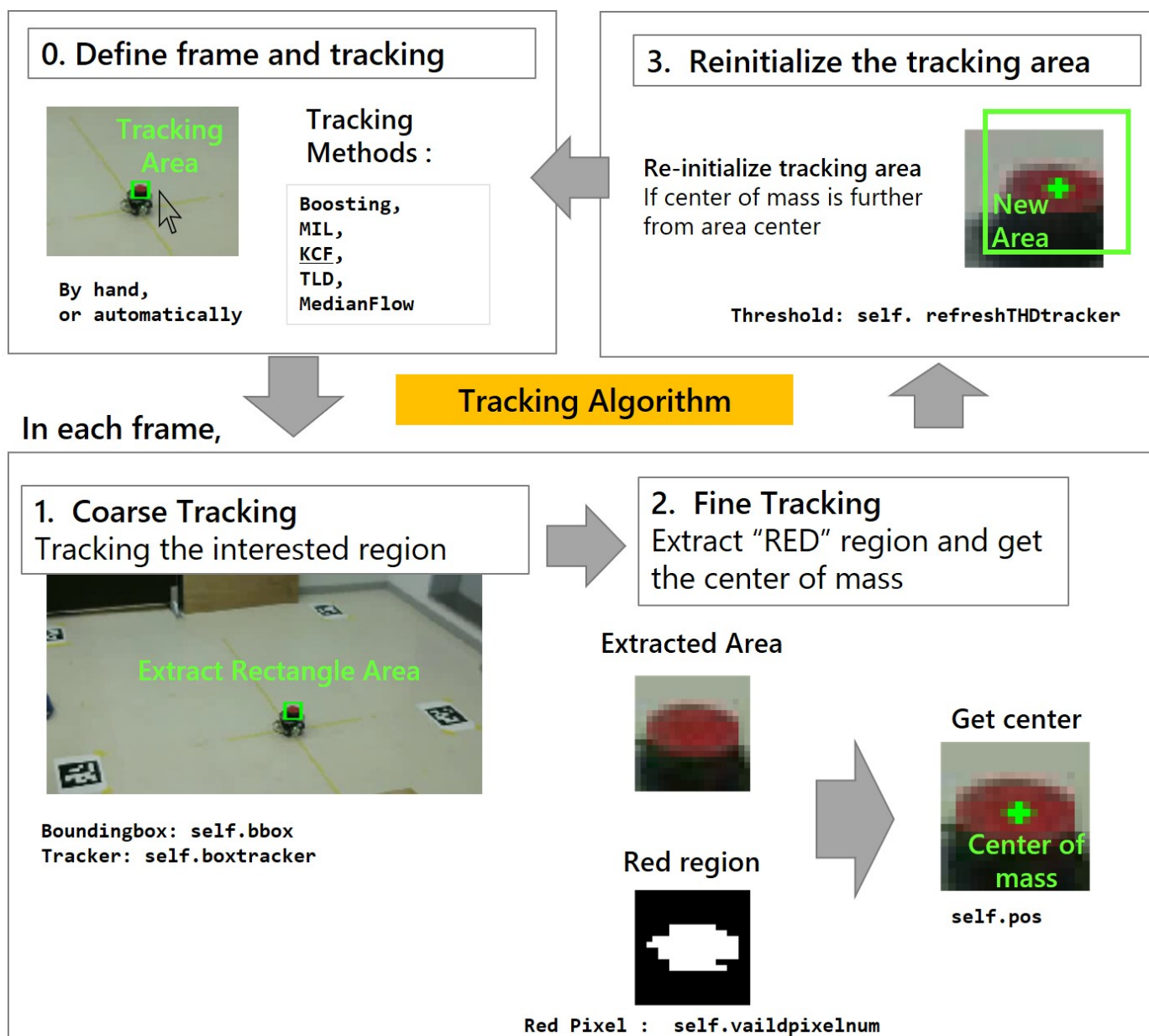


Figure 5.3 Course and Fine tracker algorithm flow

On the other hand, tracking marker with specific color is also popular in visual tracking. Colored markers allow smaller and simpler layout and enable faster tracking so that it is more suitable for smaller robot with realtime tracking. The problem is that the color based tracking has more disturbance thus we need some works to get rid of it.

5.2.2 Coarse and Fine tracking system

We propose two-step tracking system called coarse and fine tracking. As shown in Fig. 5.3, template tracking method was used to track rough bounding box including the red marker, then center of mass of red part is extracted to calculate fine marker position.

Since these template tracking method often cause a drifting problem, template recalculation is held when the distance between a center of the boundingbox and the estimated one exceed a certain value.

Fig. 5.3 shows 2D tracking results with only coarse colored center, template tracking, and the proposed coarse and fine method. Apparently, the proposed two-step estimation improves both robustness and accuracy.

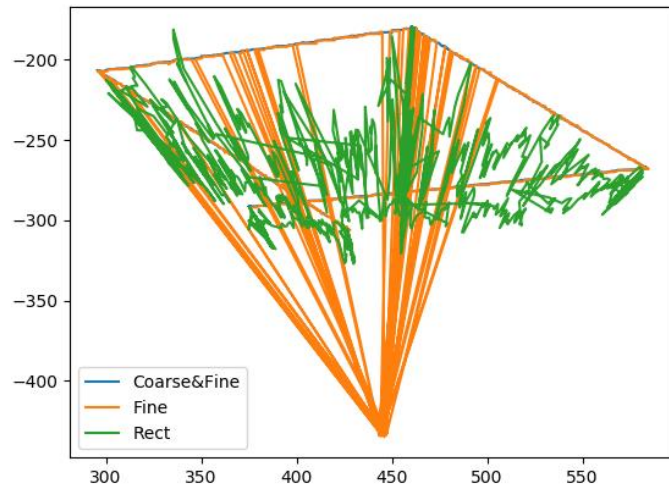


Figure 5.4 Course and Fine tracking comparison.

5.3 Position estimation via height constraint

The information obtained from the image is two-dimensional and ambiguity remains in the depth direction. There are methods such as increasing viewpoints and adding constraint conditions as means to resolve ambiguity, but it is possible to add height constraints to vehicles.

5.3.1 Camera projection model

Assume an object is at $\mathbf{X} = [X, Y, Z]^\top$ in world coordinate and a camera for observation is at position \mathbf{t}_c and attitude \mathbf{R}_c . Then, the relationship between the object coordinate in the observed image points $\mathbf{x} = [x, y]^\top$ and these global coordinates can be written as follows:

$$s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{K} [\mathbf{R} \quad \mathbf{t}] \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \quad (5.1)$$

Also, \mathbf{R}, \mathbf{t} means camera pose described on the camera coordinate and it can convert to the global coordinates with this equation: $\mathbf{R} = \mathbf{R}_c^\top, \mathbf{t} = -\mathbf{R}_c^\top \mathbf{t}_c$.

, while s means scaling value, which is equal to the inverse of the depth of the object from camera coordinate cZ_o . \mathbf{K} , a 3x3 camera matrix, contains focal lengths and image center information.

$$\mathbf{K} = \begin{bmatrix} f & fs & u_0 \\ 0 & fr & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.2)$$

We can combine these matrices to 3x4 matrix $\mathbf{P} = \mathbf{K} [\mathbf{R} \quad \mathbf{t}]$ and this \mathbf{P} is called as perspective matrix.

$$\mathbf{P} = \mathbf{K} [\mathbf{R} \quad \mathbf{t}] \quad (5.3)$$



Figure 5.5 Depth image of the experimental fields. Lighter area means further distance.

In Eq. (5.1), the problem of estimating the pose of a camera from a combination of known 3D points \mathbf{X} and imaging points \mathbf{x} is called a PnP problem. We used OpenCV PnP solver to get the ceiling camera pose from four known markers in Fig. 5.1.

5.3.2 Solve position estimation via height constraint

If the indoor floor is sufficiently flat and the vehicle robot does not jump, the marker mounted on the ground vehicle can be regarded as having a constant height constraint.

Applying these assumptions to the projection equation, the relationship between the known in-image coordinates (x, y) and the height constraint condition $Z = h$ and the camera pose becomes,

$$s \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = [\mathbf{p}_1 \quad \mathbf{p}_2 \quad \mathbf{p}_3 \quad \mathbf{p}_4] \begin{pmatrix} X \\ Y \\ h \\ 1 \end{pmatrix} \quad (5.4)$$

$$\text{while, } \mathbf{p}_i (i = 1, 2, 3, 4) \in \mathbb{R}^{3 \times 1} \quad (5.5)$$

Then, reorganizing this equation with unknown the parameters s, X, Y gives the following equation.

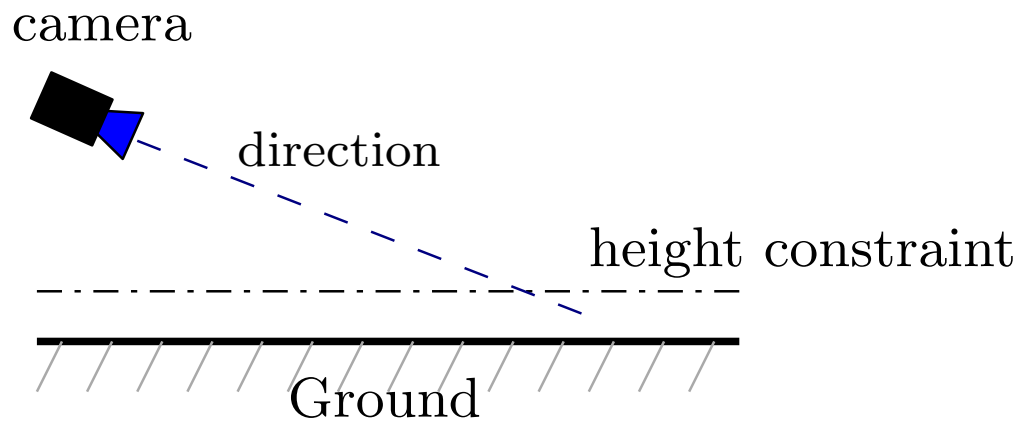
$$\begin{bmatrix} -x & \mathbf{p}_1 & \mathbf{p}_2 & \mathbf{p}_3 + h\mathbf{p}_4 \\ -y & & & \\ -1 & & & \end{bmatrix} \begin{pmatrix} s \\ X \\ Y \\ 1 \end{pmatrix} = \mathbf{0} \quad (5.6)$$

Therefore, this homogeneous equation with the form $\mathbf{A}\mathbf{x} = \mathbf{0}$ can be solved with taking a vector corresponding to the minimum singular value of the matrix \mathbf{A} represented as the left side of the Eq. (5.6).

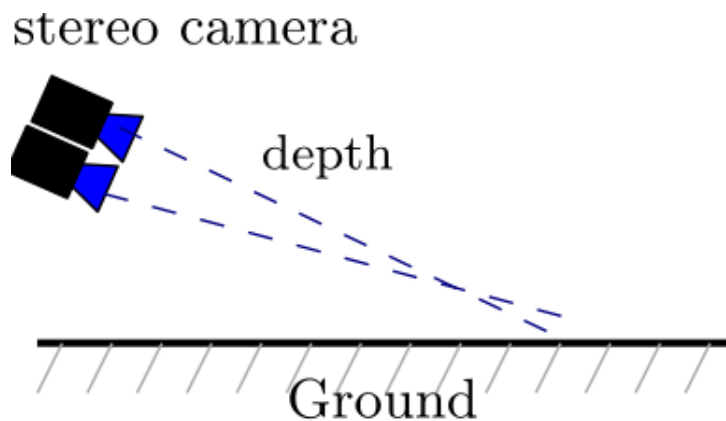
5.3.3 3D reconstruction from multiple view

For the comparison, we also used depth sensing based position estimation. With the two more ceiling cameras, we can get the depth information from stereo disparity shown in Fig. 5.5.

With the projection model in Eq. (5.1) and depth information cZ_o from stereo camera, marker position in the world coordinate $[{}^wX_o, {}^wY_o, {}^wZ_o]^\top$ has relation ship expressed as:



(a) Proposed height constraints based position estimation.



(b) Stereo depth based position estimation.

Figure 5.6 Comparison of the methods used in evaluation.

$${}^c Z_o \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathbf{K} \begin{bmatrix} \mathbf{R} & \mathbf{t} \end{bmatrix} \begin{pmatrix} {}^w X_o \\ {}^w Y_o \\ {}^w Z_o \\ 1 \end{pmatrix} \quad (5.7)$$

Rearrange this equation with regard to $[{}^w X_o, {}^w Y_o, {}^w Z_o]^\top$ then we get

$$\begin{pmatrix} {}^w X_o \\ {}^w Y_o \\ {}^w Z_o \end{pmatrix} = {}^w \mathbf{R}_c \mathbf{K}^{-1} {}^c Z_o \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} + {}^w \mathbf{t}_c. \quad (5.8)$$

Thus, the object position can be calculated with Eq. (5.8).

5.4 Experimental evaluation

We conducted experimental evaluation to compare the proposed height constraint based method and depth information based method. As shown in Fig. 5.7, a linear stage with a liner encoder was set for the groundtruth, then a red marker for tracking was set on the moving stage.

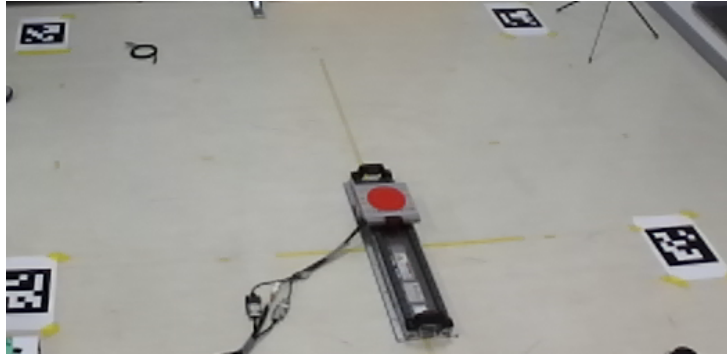


Figure 5.7 Evaluation setup. A linear stage with a red marker set along to X axis. The linear encoder was used for groundtruth, then images for estimation were captured from a stereo camera.

We had two experiments: one was along with X axis shown in Fig. 5.7 and the other was with Y axis motion. In each experiment, the linear stage was moved with sine wave position reference and a stereo camera recorded video to calculate position later.

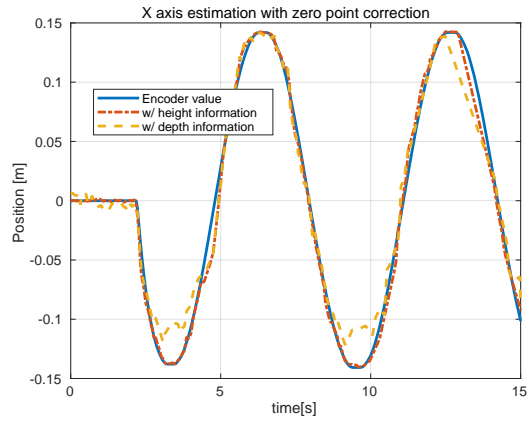
Fig. 5.8 shows X axis motion estimation results and Fig. 5.9 shows Y axis ones. In the X axis estimation, the proposed method has 1.9mm mean error with 7mm standard deviation and it is better than depth based method with 4.1mm error and 14mm standard deviation. In the Y axis estimation, the proposed method has 0.25mm mean error with 3.9mm standard deviation and it is better than depth based method with 0.45mm error and 3.1mm standard deviation.

As a discussion, the difference of accuracy between this two axes is highly related with the camera direction. The camera observation direction is almost along with X axis so that we can say estimation along the camera depth direction is harder the other direction.

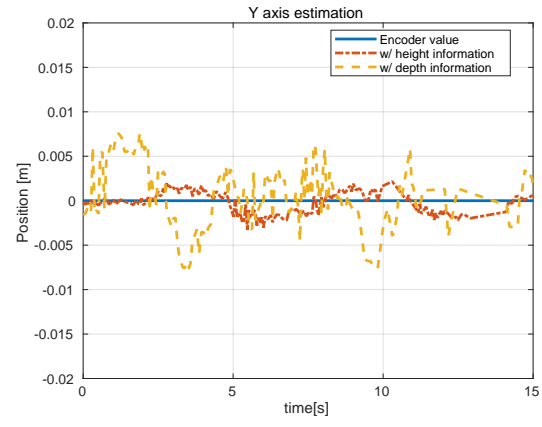
5.5 Conclusion

We constructed indoor experiment fields with an absolute positioning sensor using the ceiling camera. Assuming the constant height of ground vehicles, we can get a more efficient and accurate position estimation with the proposed method.

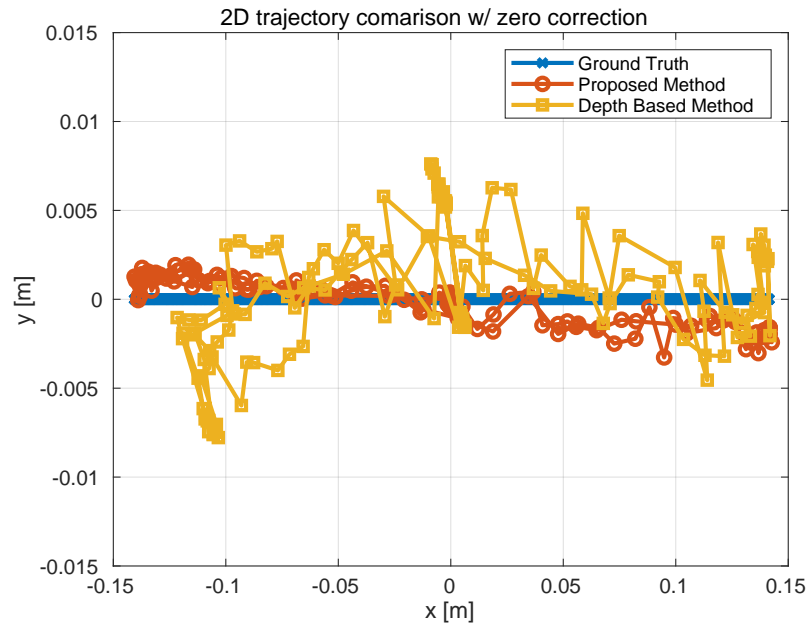
Compared with the results in [47], which uses 20cm marker sets to result in mm-order localization with about 1.5m range, our approach can achieve the similar accuracy with 8cm diameter circle marker and over 3m range.



(a) X axis motion estimation.

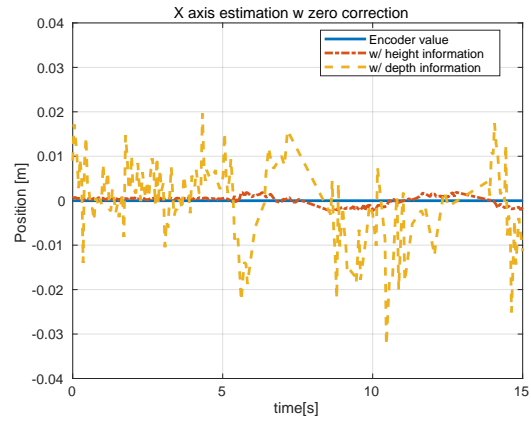


(b) Y axis motion estimation.

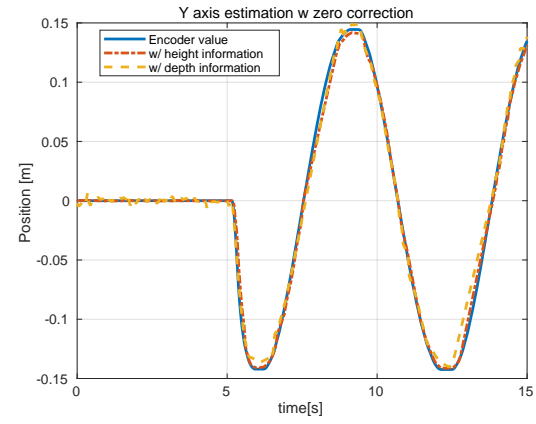


(c) 2D motion estimation.

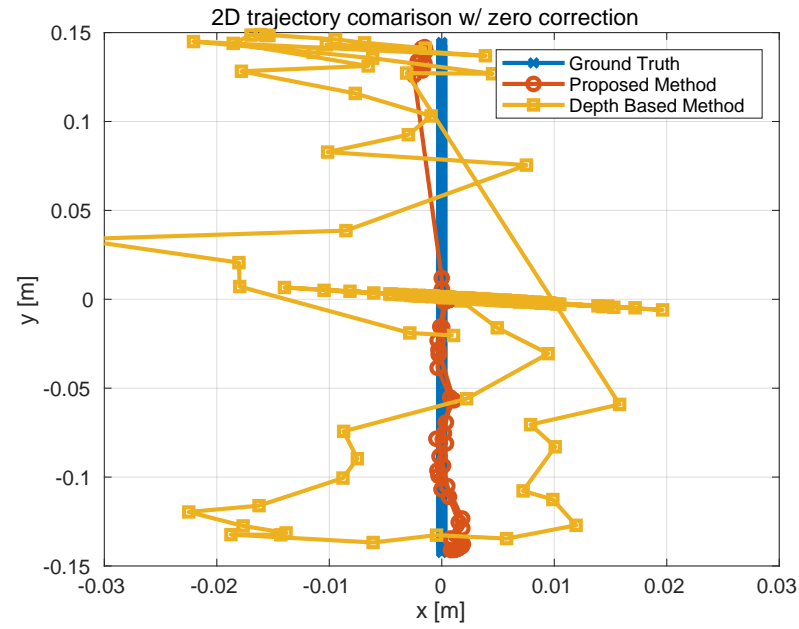
Figure 5.8 Evaluation with X axis motion.



(a) X axis motion estimation.



(b) Y axis motion estimation.



(c) 2D motion estimation.

Figure 5.9 Evaluation with Y axis motion.

Chapter 6

Switching Observer based Motion Estimation via Stereo Disparity and Monocular Scaling Sensor Fusion

6.1 Motivation

This chapter focuses on the multi observation based depth reconstruction method. Stereo camera, a system to measure the depth from a pair of cameras, is also very popular in position based visual servo [49].

6.1.1 Stereo measurement

An object depth information can be estimated with more than two observations from different perspectives. Fig. 6.1 shows the simplest principle of a stereo measurement. In a stereo vision system, an object in the distance Z is converted to a horizontal error between the left and right images. This position error, which is called as a disparity D , is inversely proportional to the object distance Z like below:

$$D = \frac{bf}{Z} \quad (6.1)$$

while b and f mean fixed parameters representing a baseline and a focal length. b and f are can be estimated in a calibration step, depth Z measurement is equivalent to disparity D measurement.

Finding corresponding parts in left and right images is one of the main difficulty in the stereo depth measurement [50]. Sum of absolute difference between the certain window regions are used in this paper for the faster estimation acquisition.

Stereo range limitation

The disparity estimation is restricted between a minimum disparity D_{min} and a maximum disparity D_{max} due to its software and hardware limitations. For example, minimum disparity is around $1pix$ because of the pixel quantization. On the other hand, disparity search is limited to certain upper bound value D_{max} so that it can reduce the mismatch probability and save computation time. In addition, the left end of left image and the right end of right image have no overlap area so that there is no information.

In our case, minimum disparity is around $D_{min} = 1pix$ and maximum disparity is set as $D_{max} = 80pix$:

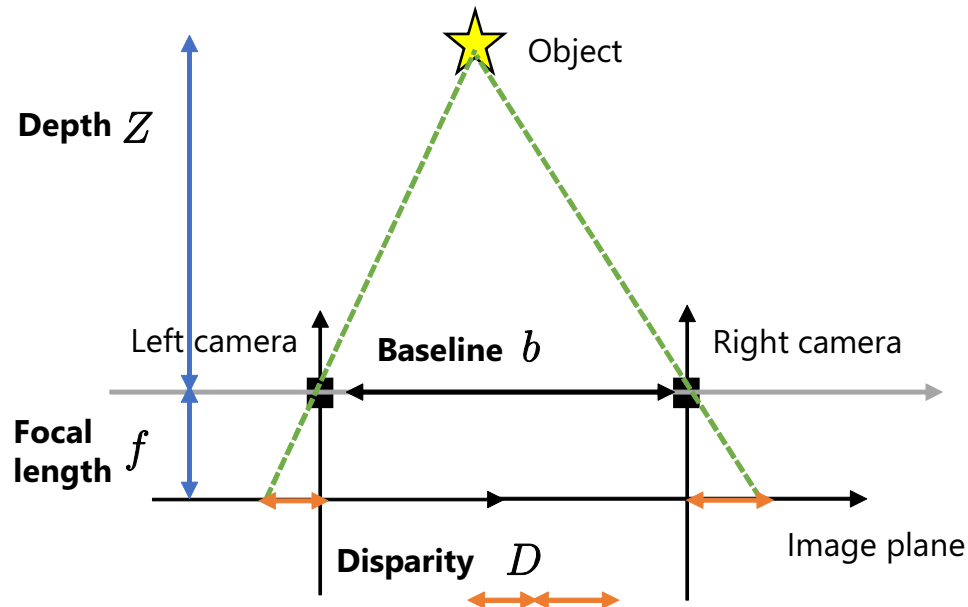


Figure 6.1 The principle of a stereo camera.

so depth range can be calculated as $Z_{max} = 25.4[m]$ and $Z_{min} = 0.4$ based on Eq. (6.1) and Table. 6.1.

6.1.2 Monocular measurement

There were many researches about absolute or relative depth estimation from single camera because a monocular camera is now more common and easy equipment for the industry. Monocular camera based depth estimations need some comparison image: In an active stereo, current image is compared with the previous image.

Even the state-of-the-art deep convolutional neural network based depth estimation [51] using the advance information about objects shape.

In this paper the template tracking are used to estimate the relative scaling from the template. If the template is planer and orthogonal to light axis of the camera, the scaling S and depth Z have a inverse propositional relationship.

$$S = \frac{Z_0}{Z} \quad (6.2)$$

where, Z_0 denotes the depth when the object size in the image become equal to that of the template and it is constant variant.

6.2 Algorithm flow

This section shows more detail about the proposed object tracking and depth estimation method.

6.2.1 Template extraction based on machine learning

As for the template extraction, convolutional neural network based object extraction method is used. Fig. 6.6 shows the left camera view with detected object. For the object detection, YOLO (You Look Only Once) [52] algorithm implemented on the ROS (Robot-Operation-System) [53] is used and it returns

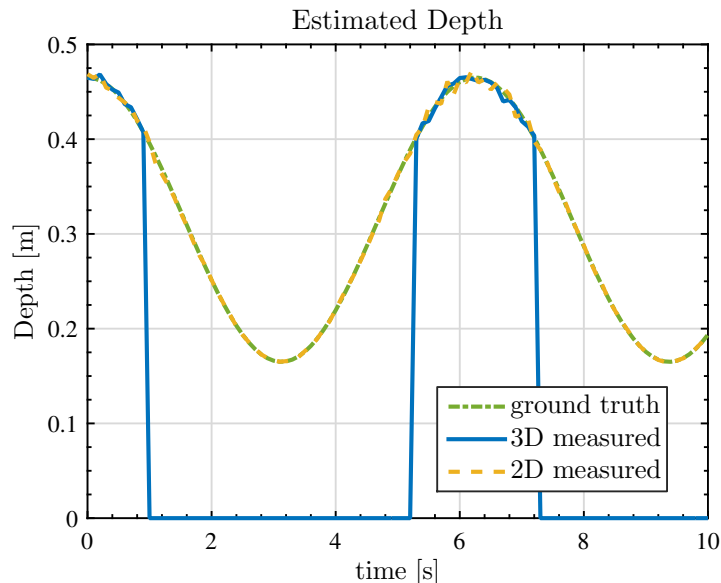


Figure 6.2 Depth estimation results from each raw measurements.

bounding box which represent object position and its classification result. Thus, the template image for tracking is extracted automatically before the experiment.

6.2.2 Template tracking using SURF [2] feature and scale extraction

Then, traditional template tracking with SURF (Speeded Up Robust Features) [2] is applied. Finally, we get a homography matrix with robust parameter estimation using RANSAC (Random Sample Consensus) [26].

The scaling parameter from a monocular vision in this chapter is defined as inverse depth:

$$\lambda = \frac{1}{Z_0} \quad (6.3)$$

where, Z_0 is the depth in the template image.

The approximate scaling λ' can be estimated from the estimated 3×3 homography matrix \mathbf{H} :

$$\lambda' = \left(\frac{h_{11}^2 + h_{12}^2 + h_{21}^2 + h_{22}^2}{2} \right)^{\frac{1}{2}} \quad (6.4)$$

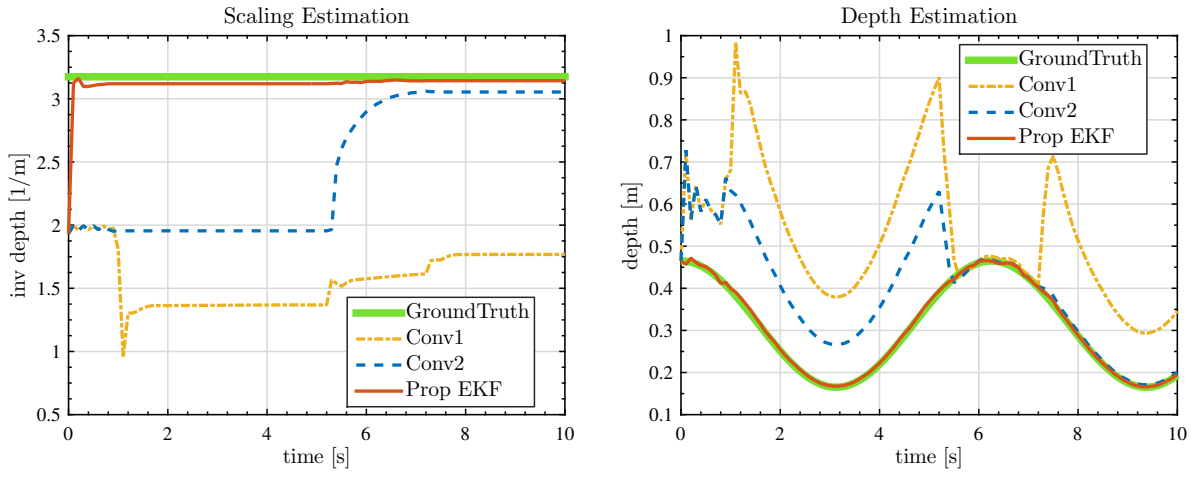
while, h_{ij} means i th row and j th column ingredient of \mathbf{H} . To obtain more precise value, the singular value decomposition based estimation in A can be used.

After this tracking, we obtain current template scaling and its new bounding box area.

6.2.3 Stereo disparity measurement

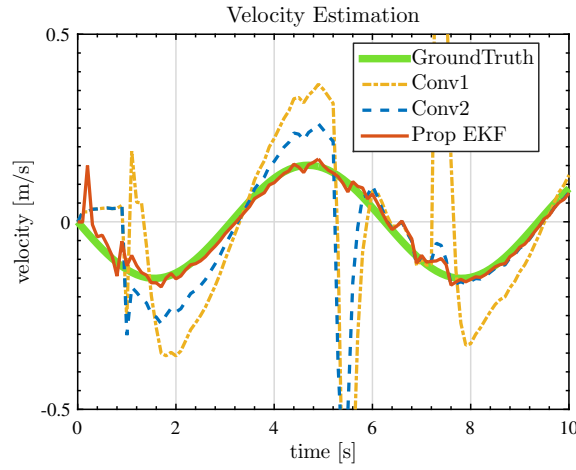
The stereo measurement is obtained after the object position in found in the image plane. Since a raw disparity measurement often contains invalid value and outliers, preprocessing to exclude these undesired value is applied.

A histogram of disparity can easily exclude these outliers, and the measured disparity D is calculated as a average value in a mode bin of the histogram. This process also give us additional information if

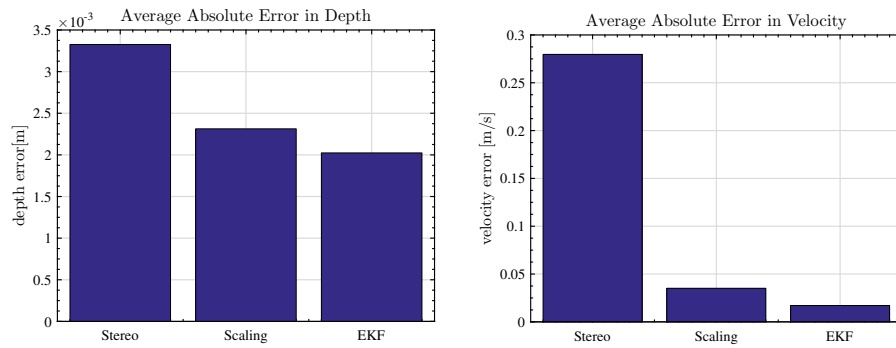


(a) Scaling λ estimation result

(b) Depth Z estimation result



(c) Velocity \dot{Z} estimation result



(d) Average estimation error of Z .

(e) Average estimation error of \dot{Z} .

Figure 6.3 Simulation Results with Sin Wave Motion.

this stereo estimation is reliable. Define an inlier rate η as following equation:

$$\eta = \frac{\mathbf{I}_{Inlier}}{\mathbf{I}_{Total}} \quad (6.5)$$

where \mathbf{I}_{Total} denotes the number of pixels belonging to estimated rectangle and \mathbf{I}_{Inlier} denotes that of inlier pixels after histogram processing.

Algorithm 1 EKF based stereo and monocular information fusion algorithm

Require: Initial state variable estimation \mathbf{X}_{init} and measurement data with timestamps

- 1: **for** Each measurement with a timestamp t_n **do**
 - 2: Calculate sampling interval from last step $\Delta t = t_n - t_{n-1}$
 - 3: Calculate \mathbf{A}, \mathbf{B} matrix in Eq. (6.7)
 - 4: State and covariance prediction : $\hat{\mathbf{X}}_n \leftarrow \mathbf{A}\mathbf{X}_{n-1}, \hat{\mathbf{P}}_n \leftarrow \mathbf{A}\mathbf{P}_{n-1}\mathbf{A}^\top + \mathbf{Q}$
 - 5: **if** η in Eq. (6.5) is lower than threshold **then**
 - 6: Set stereo measurement covariance $\sigma_{\omega_{h1}}^2 = \infty$
 - 7: Replace the jacobian of monocular measurement H_2 with H'_2 in Eq. (6.12)
 - 8: **end if**
 - 9: Kalman gain calculation:
 $K_1 \leftarrow \hat{\mathbf{P}}_n H_1^\top (H_1 \hat{\mathbf{P}}_n H_1^\top + R_1)^{-1}$
 $K_2 \leftarrow \hat{\mathbf{P}}_n H_2'^\top (H_2' \hat{\mathbf{P}}_n H_2'^\top + R_2)^{-1}$
 - 10: State and covariance update with disparity and scaling information:
 $\mathbf{X}_n \leftarrow \hat{\mathbf{X}}_n + (K_1 \quad K_2) \begin{pmatrix} D_n - h_1(\hat{\mathbf{X}}_n) \\ \frac{1}{S_n} - h_2(\hat{\mathbf{X}}_n) \end{pmatrix}$
 $\mathbf{P}_n \leftarrow \left(\text{eye}(3) - (K_1 \quad K_2) \begin{pmatrix} H_1 \\ H_2' \end{pmatrix} \right) \hat{\mathbf{P}}_n$
 - 11: **return** $\mathbf{X}_n, \mathbf{P}_n$
 - 12: **end for**
-

6.3 Sensor fusion filtering design for stereo and monocular sensing mixture

In this section EKF based sensor fusion technique is applied to object depth and velocity estimation. The state-space model is derived considering the past research on the sensor fusion of inertial sensors and monocular vision [54].

6.3.1 State-space model

A state-space model is needed for formulating the depth estimation problem in a Kalman filter framework. A motion model for object is considered as a simple random walk process and a state variable is chosen as $\mathbf{X} = \begin{pmatrix} \lambda & Z & \dot{Z} \end{pmatrix}^\top$: it contains scaling λ , depth Z and its velocity \dot{Z} .

Then, the continuous state-space equation can be written as following:

$$\frac{d}{dt} \begin{pmatrix} \lambda \\ Z \\ \dot{Z} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ Z \\ \dot{Z} \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \omega \quad (6.6)$$

In Eq. (6.6), ω denotes a gaussian random walk of acceleration with 0 average and σ_ω^2 covariance; from now on, we express this gaussian distribution as $\omega \sim \mathcal{N}(0, \sigma_\omega^2)$.

Table 6.1 Camera and motion parameter in simulation

Camera parameter		Motion parameter	
Baseline b	0.0654 [m]	Initial Depth Z_0	0.315 [m]
Focal length f	396 [pix/m]	Amplitude α	0.15 [m]
Max disparity	65 [pix]	Frequency ϕ	0.16 [Hz]

Table 6.2 EKF parameter in simulation

Noise parameter	
Stereo Disparity Noise σ_{h1}^*	0.5 [pix]
Inverse Scaling Noise σ_{h2}^*	$Z/0.0212$ (variable)
System Noise	0
EKF parameter	
Stereo Disparity Noise σ_{h1}	0.5 [pix]
Inverse Scaling Noise σ_{h2}	0.01
System Noise / Random walk σ_ω	0.1

Then, we have a discrete state-equation Eq. (6.7) by discretization of Eq. (6.6) with sampling Δt .

$$\begin{pmatrix} \lambda_{n+1} \\ Z_{n+1} \\ \dot{Z}_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \lambda_n \\ Z_n \\ \dot{Z}_n \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{\Delta t^2}{2} \\ \Delta t \end{pmatrix} \omega \quad (6.7)$$

In this paper, we use the stereo disparity D_n and the inverse of monocular scaling S_n^{-1} and those measurement equations, can be written as following:

$$D_n = h_1(\lambda_n, Z_n, \dot{Z}_n) = \frac{bf}{Z_n} + \omega_{h1} \quad (6.8)$$

$$S_n^{-1} = h_2(\lambda_n, Z_n, \dot{Z}_n) = \lambda_n Z_n + \omega_{h2} \quad (6.9)$$

where, $\omega_{h1} \sim \mathcal{N}(0, \sigma_{\omega_{h1}}^2)$ and $\omega_{h2} \sim \mathcal{N}(0, \sigma_{\omega_{h2}}^2)$ denotes measurement error covariance.

Since these equations are non-linear, so EKF uses 1st order of Taylor approximation around the current state \mathbf{X}_n ; Eq. (6.10) and Eq. (6.11) are called as jacobian.

$$H_1 = \frac{dh_1}{d\mathbf{X}_n} = \begin{pmatrix} 0 & -\frac{bf}{Z_n^2} & 0 \end{pmatrix} \quad (6.10)$$

$$H_2 = \frac{dh_2}{d\mathbf{X}_n} = \begin{pmatrix} Z_n & \lambda_n & 0 \end{pmatrix} \quad (6.11)$$

6.3.2 EKF estimation flow

Actual EKF estimation flow is shown in Algorithm 1. Since image processing and camera sampling often do not have a constant time interval, each step requires recalculation of the discrete state-equation of Eq. (6.7).

The difference between ordinary EKF algorithm and proposed one in Algorithm 1 is that proposed one has switching mechanism based on inlier calculated in Eq. (6.5). In the actual estimation, the jacobian

of monocular estimation in Eq. (6.11) is replaced by following equation:

$$H'_2 = \frac{dh_2}{d\mathbf{X}_n} = \begin{cases} \begin{pmatrix} 0 & \lambda_n & 0 \end{pmatrix} & \text{(stereo not available)} \\ \begin{pmatrix} Z_n & 0 & 0 \end{pmatrix} & \text{(stereo available)} \end{cases} \quad (6.12)$$

When there is only scaling data can be obtained, Eq. (6.11) becomes the ill-posed problem that estimates two state variables from a mere single value observation. Therefore, in that situation, we give up updating scaling parameter λ and thought it as constant value. On the other hand, when the stereo estimation is available, we change the Eq. (6.11) as Eq. (6.12) so that it will faster the convergence of scaling factor λ .

The covariance matrices Q, R_1, R_2 in the Algorithm 1 is respectively written as Eq. (6.13).

$$Q = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} \\ 0 & \frac{\Delta t^3}{2} & \Delta t^2 \end{pmatrix} \sigma_\omega^2, \quad R_1 = \sigma_{\omega_{h1}}^2, \quad R_2 = \sigma_{\omega_{h2}}^2 \quad (6.13)$$

6.3.3 Convergence analysis with linearized poles

With the linearization around current estimation state, observation poles of this linearized error system model can be calculated from following equations:

$$\begin{aligned} \mathbf{e}_n &\approx (\mathbf{A} - \mathbf{K}_n \mathbf{C}_n \mathbf{A}) \mathbf{e}_{n-1} \\ \text{while, } \mathbf{C}_n &= \left. \frac{d\mathbf{h}}{d\mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_n} \end{aligned} \quad (6.14)$$

Then, the eigen values of $(\mathbf{A} - \mathbf{K}_n \mathbf{C}_n \mathbf{A})$ represents the convergence rate at that moment.

6.3.4 Linearized observer based estimation method

This part explains robust pole placement method called eigenvalue assignment.

When the eigen values of $(\mathbf{A} - \mathbf{K}_n \mathbf{C}_n \mathbf{A})$ is p_i and its eigen vector is \mathbf{v}_i , the relationships is

$$(\mathbf{A}^\top - (\mathbf{C}_n \mathbf{A})^\top \mathbf{K}_n^\top) \mathbf{v}_i = p_i \mathbf{v}_i. \quad (6.15)$$

This equations can be transformed into Eq. (6.16).

$$(p_i \mathbf{I} - \mathbf{A}^\top \quad (\mathbf{C}_n \mathbf{A})^\top) \begin{pmatrix} \mathbf{v}_i \\ \bar{\mathbf{K}}_n \mathbf{v}_i \end{pmatrix} = 0 \quad (6.16)$$

With SVD and get null space of $(p_i \mathbf{I} - \mathbf{A}^\top \quad (\mathbf{C}_n \mathbf{A})^\top)$ leads to $\begin{pmatrix} \mathbf{v}_i \\ \bar{\mathbf{K}}_{in} \mathbf{v}_i \end{pmatrix}$ in Eq. (6.16). Repeating this process for the number of the eigen value and concatenating estimated $\begin{pmatrix} \mathbf{v}_i \\ \bar{\mathbf{K}}_{in} \mathbf{v}_i \end{pmatrix}$ to acquire $\begin{pmatrix} V \\ Q \end{pmatrix}$:

$$\begin{pmatrix} V \\ Q \end{pmatrix} = \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\ \bar{\mathbf{K}}_{1n} \mathbf{v}_1 & \bar{\mathbf{K}}_{2n} \mathbf{v}_2 & \bar{\mathbf{K}}_{3n} \mathbf{v}_3 \end{pmatrix} \quad (6.17)$$

Finally, the observer gain which satisfy the constraints \mathbf{K}_n is

$$\mathbf{K}_n = QV^{-1}. \quad (6.18)$$

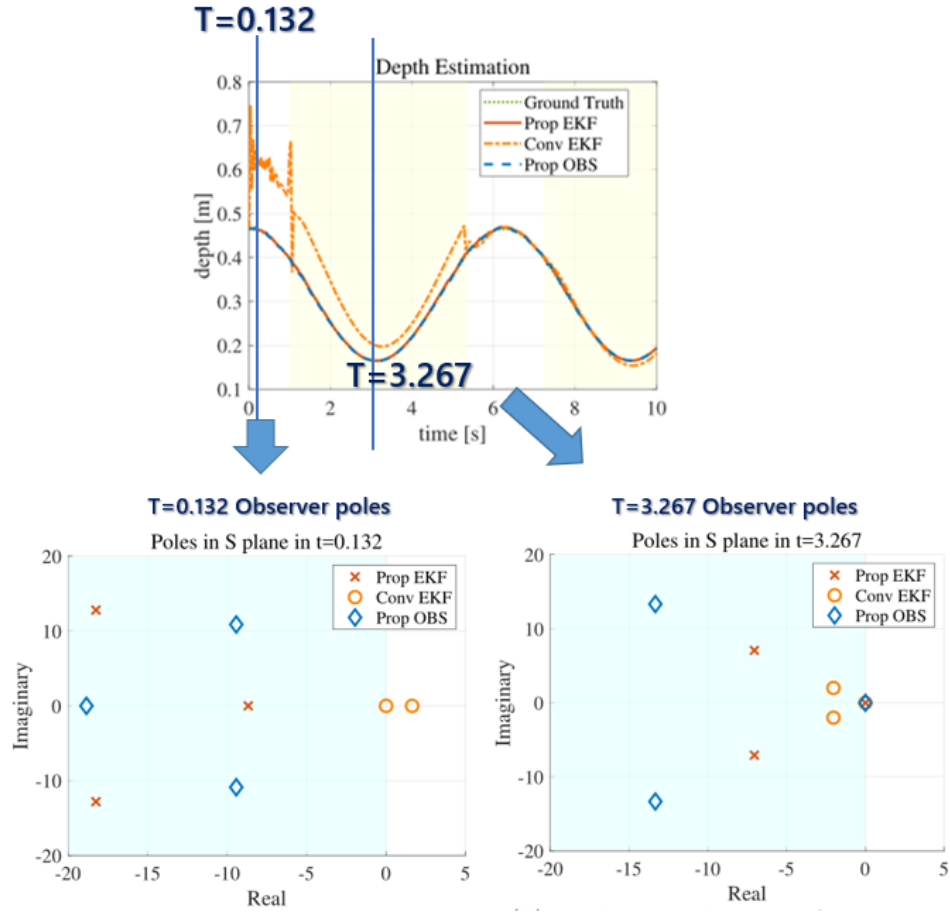


Figure 6.4 Linearized pole based analysis. The poles are shown in s plane.

6.4 Simulation

Simulations were held to show the effectiveness of the proposed switching Kalman filter system. Simulation parameters are shown in Table. 6.1. The camera and motion parameters are the same with experimental setup. In this simulation, a camera was assumed to be fixed and an object was to do sin wave motion with $Z = Z_0 + \alpha \cos 2\pi\phi t$.

Measurement noises shown in Table. 6.2 were added in this simulation and the raw depth estimations based on noised measurements are shown in Fig. 6.2. In Fig. 6.2, measurement noises become larger when the object is farther away and this imitates real measurements.

Measurement sampling was set as 100ms and stereo and monocular estimation was considered as simultaneous signals. Fig. 6.3 shows the estimation results based on the proposed method and two comparison methods. The yellow lines named Conv1 used \mathbf{H}_2 explained in Eq. (6.11) whole time, and the estimations of the scaling λ and the physical information Z did not converge to the ground truth. The blue lines named Conv2 partially used switching technique shown in Eq. (6.11) only in the absence of the stereo measurement. Fig. 6.3(a) shows the slower convergence of the scaling estimation and thus it resulted in slower estimations in Fig. 6.3(b) and Fig. 6.3(c).

These results well explains the necessity of the proposed switching systems.

Tuning parameters in our method are covariances of measurements σ_{h1}, σ_{h2} and system noise σ_ω . Generally speaking, a larger measurement covariance causes less noise and larger phase delay; and vice versa in a smaller measurement covariance. So, the upper bound of σ_{h1}, σ_{h2} should be decided from

the maximum acceptable phase delay; and the lower bound should be decided from acceptable noise magnitude.

6.5 Experiment

Simple experiments using a linear stage were held to evaluate the proposed depth estimation method.

6.5.1 Experimental setup

Fig. 6.5 shows the experimental setup. The stereo camera was fixed on the moving linear stage and a planer object was put on the wall; this problem setting is essentially the same with a static camera and a moving object.

An linear encoder on the linear stage provided ground truth of motion and disparity and scaling estimation was calculated in real-time. The state estimation were executed off-line after this real-time data processing and acquisition.

Image processing was done on NVIDIA Jetson TX2 CPU/GPU board and each program was implemented on ROS Kinetic with c++/Python on Ubuntu 16.04.

Due to hardware and software specifications, the sampling periods of sensors and processing are not fixed; ROS currently does not support a rigid real-time implementation but provides timestamps of the obtained sensor data. The processing times are roughly around 100 ms. Therefore, the discrete state-equation in the each EKF estimation step was derived using each sampling interval Δt_n based on the timestamps from ROS. Tuning parameters of the EKF was set as the same with the simulation, shown in Table. 6.1.

6.5.2 Results and discussion

Fig. 6.7 shows the one of the raw depth data containing ground truth and estimations from raw data of the stereo and monocular measurements with proposed methods. The blue lines in Fig. 6.7 and Fig. 6.8 shows its 95% confidence intervals [55] calculated from states \mathbf{X}_n and its covariance \mathbf{P}_n . In Fig. 6.7 and Fig. 6.8, the scaling parameter $\lambda = 1/Z_0$ and the relationship between linear encoder value and camera depth are manually measured. Also, invalid stereo measurements are set to 0. These results indicate a fact in the image based depth sensing: Depth measurements become more noisy with the farther object and it is the same with assumption in the simulation in the previous section.

Fig. 6.9 and Fig. 6.10 show the EKF estimation results with the sin wave motion and sudden motion. Fig. 6.9(a), Fig. 6.9(b) and Fig. 6.9(c) show the estimation results are well fit to the ground truth. Fig. 6.10 shows even some sudden motions happen, proposed method can immediately track the ground truth. Quantitative evaluations for the average error of valid data are shown in Fig. 6.9(d) and Fig. 6.9(e). The stereo measurements were much more accurate than the monocular one in our situation, so it resulted that average error within the valid data was better than the EKF estimation which used the noisy monocular information in the closer range. Also, measurements in the disparity were more accurate in this experiment than simulation because we averaged disparities in the bounding box obtained from a monocular template tracking.

Thus, this results show that the better depth measurement can be achieved with the fusion of stereo



Figure 6.5 Experimental Setup



Figure 6.6 Detection with YOLO

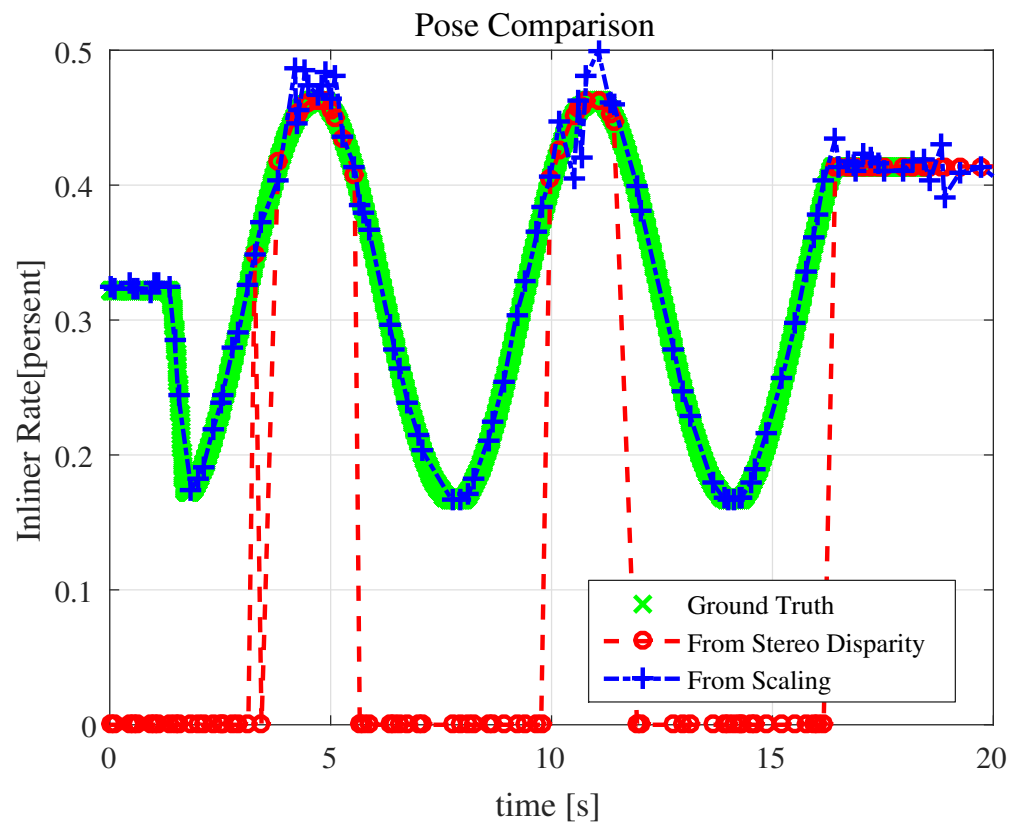


Figure 6.7 Position data from linear encoder, stereo disparity and scaling. (with a sin wave motion)

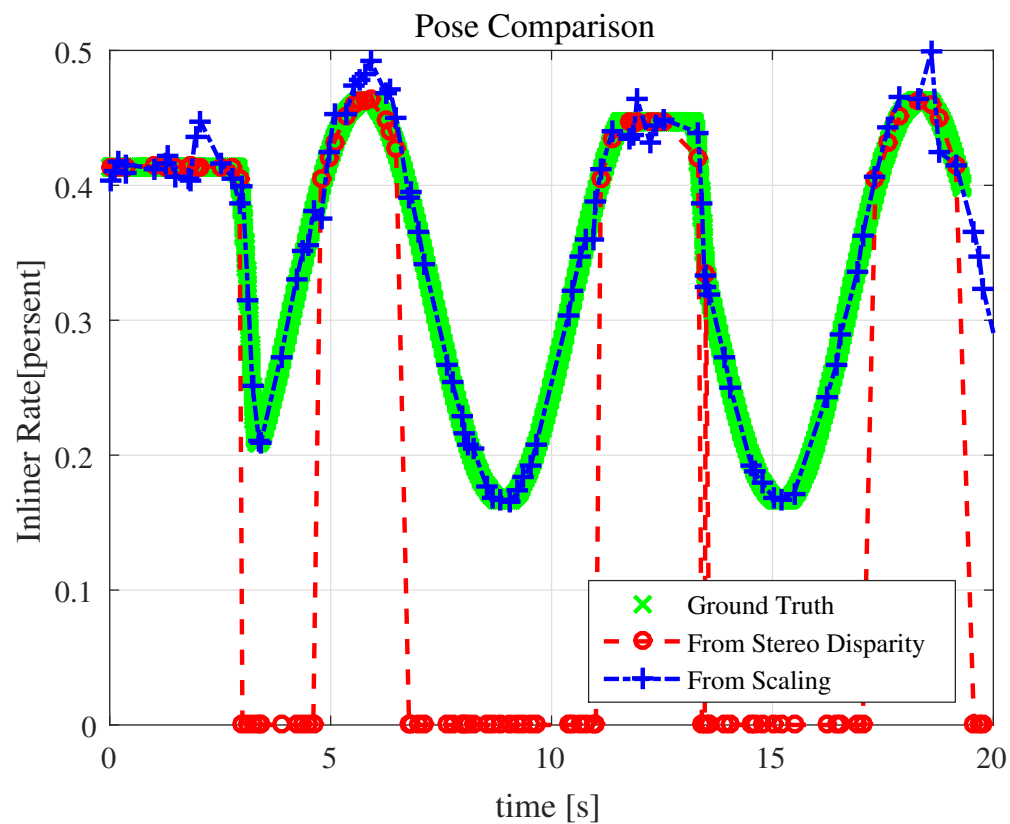
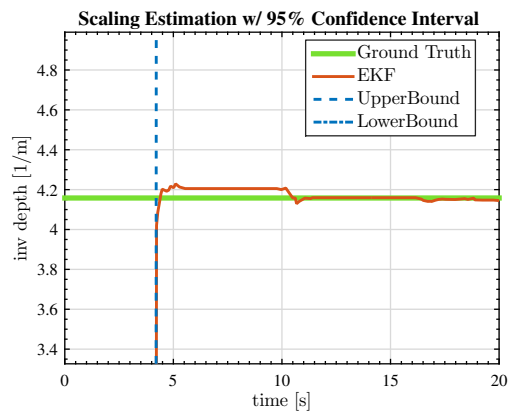
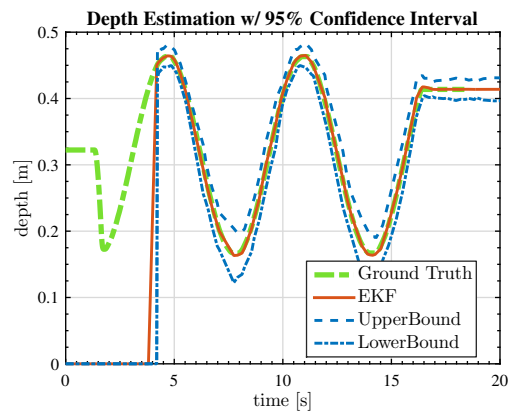


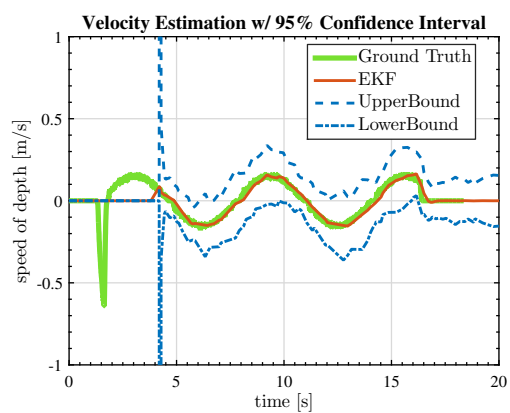
Figure 6.8 Position data from linear encoder, stereo disparity and scaling. (with a sudden motion)



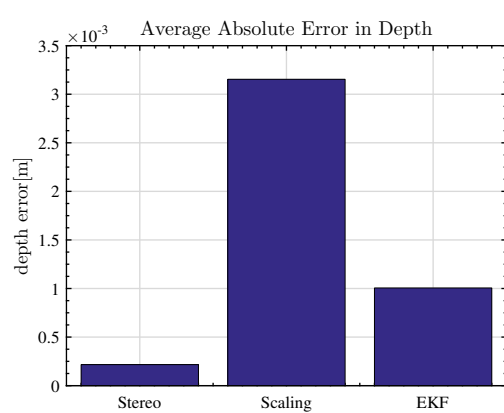
(a) Scaling λ estimation result



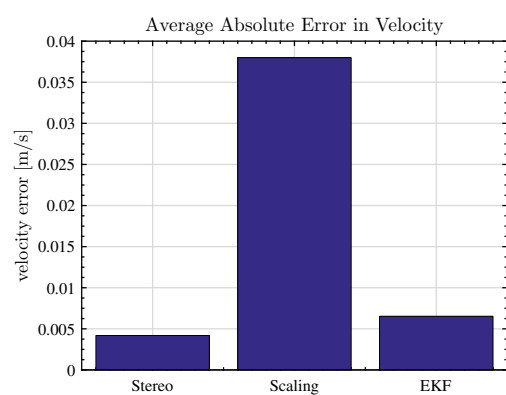
(b) Depth Z estimation result



(c) Velocity \dot{Z} estimation result

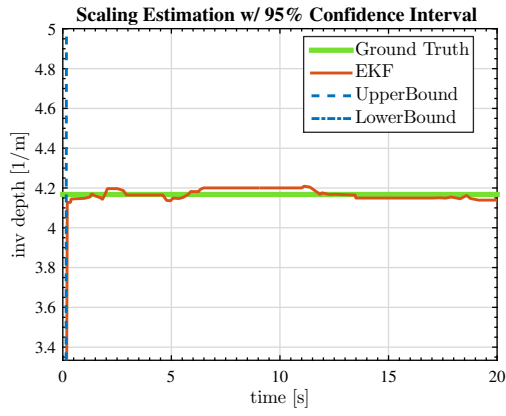


(d) Average estimation error of Z .

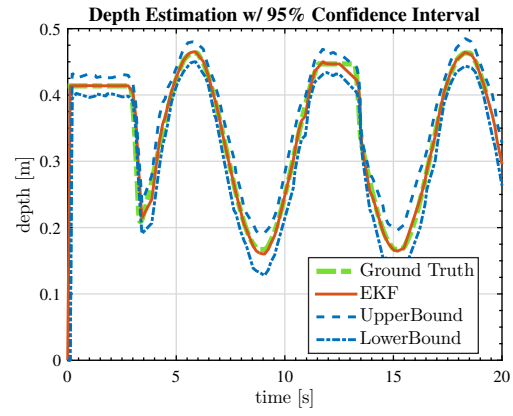


(e) Average estimation error of \dot{Z} .

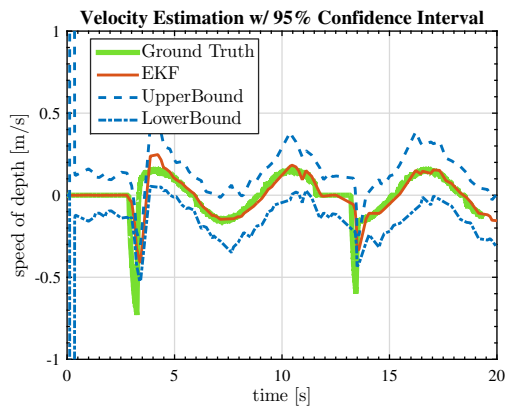
Figure 6.9 Experiment results with a sin wave motion.



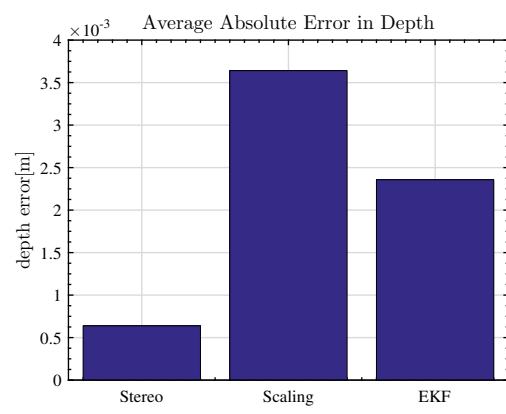
(a) Scaling λ estimation result



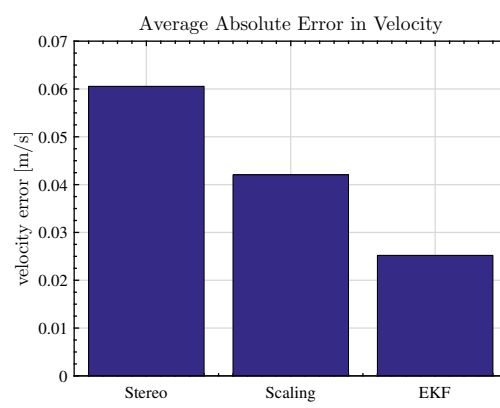
(b) Depth Z estimation result



(c) Velocity \dot{Z} estimation result



(d) Average estimation error of Z .



(e) Average estimation error of \dot{Z} .

Figure 6.10 Experiment results with a sudden motion.

and monocular estimation.

6.6 Conclusion

In this paper, we proposed the sensor fusion technique with a stereo camera using both a stereo disparity and a monocular scaling measurement. A random walk based state-space model was applied to the EKF sensor fusion approach and the jacobian switching method was applied to the proposed method. The simulation results shows its effectiveness than other normal method, and the experimental results show that it still provides good estimations in the actual systems.

These results can improve such as inter-vehicular distance control or soft landing/approaching to the planer surface with a robot or UAV(Unmanned Aerial Vehicle).

We will continue to work on;

1. Convergence analysis of the proposed switching Kalman filter and its tuning.
2. Add another motion parameter to estimate such as yaw rotation of the object.
3. Application to distance control systems.

Chapter 7

Sensor Fusion Tuning toward Vision based Relative Position Control

7.1 Motivation

Chapter 6 results shows the sensor fusion technique to acquire faster and fewer noise estimation. However, the looking-good estimation is not necessary means a good estimate for the visual servo control system.

This chapter outlines one of the distance information based control applications used in adaptive cruise control (ACC) as a part of an autonomous driving system. Starting with the popular ACC control method and its parameter introduction, we finally evaluate the sensor fusion technique with a certain control system.

7.2 Problem statement in ACC

The ACC aims to follow without delay by measuring the relative distance and speed of the preceding vehicle with a sensor and controlling the distance.

One of the merits of ACC is that it is possible to drive with good fuel efficiency by reducing air resistance by keeping automatic control while maintaining a certain distance (4 – 10m) close enough. It can also be expected to increase traffic capacity by maintaining a close inter-vehicle distance.

Fig. 7.1 shows the assumption in ACC.

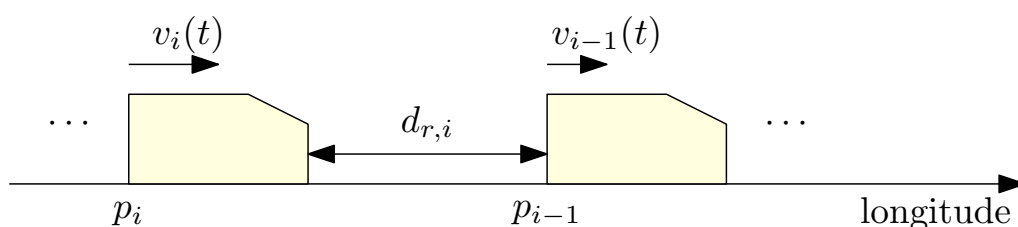


Figure 7.1 The concept of ACC and variable declaration.

The inter-vehicle distance reference of the i th vehicle $d_{ref,i}(t)$ should be proportional to the vehicle speed, which is described as

$$d_{ref,i}(t) = r_i + hv_i(t) \quad (7.1)$$

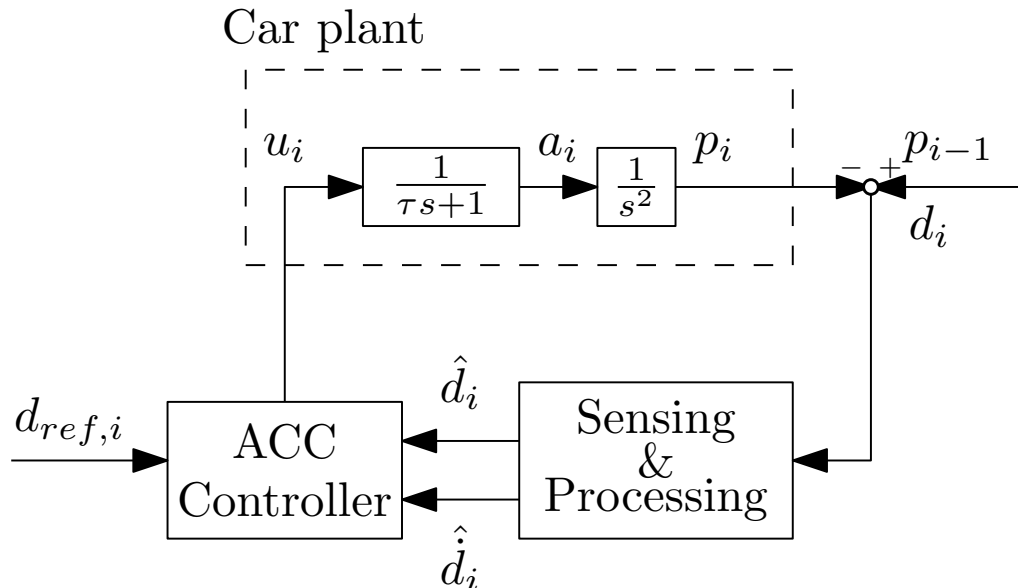


Figure 7.2 The block diagram of the ACC system.

Table 7.1 Feedback gain at each categories [6].

Group	F_1	F_2
A	0.025	0.41
B	0.030	0.30
C	0.075	0.25

in [56]. r_i is a constant called standstill distance, and h is a constant called Time headway or inter-vehicle time. The inter-vehicle time in commercial car is often set to 1.3 – 2.4 seconds [57].

The control input u_i is treated as an acceleration command value, and the car follows the given acceleration command value with a model of a temporary delay system with a time constant τ .

$$\dot{a}_i = \frac{1}{\tau}(u_i - a_i) \quad (7.2)$$

It is often estimated that τ is 0.2 seconds for general passenger cars and 0.5 seconds for large vehicles. Due to the recent progress in a driving force control on EV with in-wheel-motors [58], this τ could be faster in the future autonomous driving system.

The ACC controller has many variation such as PID [59] [60] but in this section we discuss the state feedback type approach to evaluate the results in chapter 6.

The ACC controller for determining the target acceleration u_i uses a speed and distance error multiplied by a proportional gain.

$$u_i = F_1(d_i - d_{ref,i}) + F_2(v_{i-1} - v_i) \quad (7.3)$$

The feedback parameter decision in commercial car at [57] can be classified into 3 categories, and Table. 7.1 indicates the each controller parameter.

7.2.1 String stability in ACC

The inter-vehicle distances in relative position control should be also stable in ACC. The stability goal for inter-vehicles dynamics is called string stability [57] [61]. When these transfer functions are considered

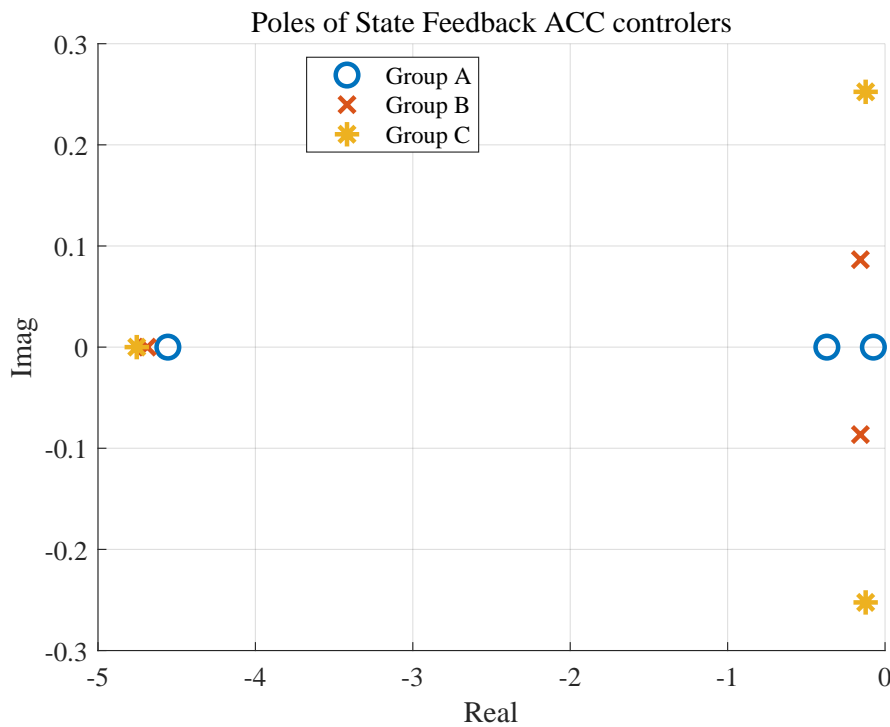


Figure 7.3 Pole positions of the continuous state-feedback controller of ACC with Eq. (7.8) parameters shown Table. 7.1.

when the i th inter-vehicle distance is expressed as d_i and the $i + 1$ th inter-vehicle distance is expressed as d_{i+1} ,

$$G(s) = \frac{D_{i+1}(s)}{D_i(s)} \quad (7.4)$$

If this is smaller than 1 in all frequency bands, the variation in the front-to-vehicle distance gradually attenuates as it goes backward, so stable platooning is maintained.

On the other hand, there are subspecies [6], such as those that think that speed fluctuations are transmitted.

According to the research in the above, none of the current vehicle control parameters are likely to meet this string stability. In particular, the feedback gain of the relative speed is low, and it is considered to prevent the influence of the relative speed error measured by the in-vehicle sensor and the uncomfortable feeling of acceleration / deceleration when using ACC.

Nowadays, Cooperative Adaptive Cruise Control (CACC) in which vehicles communicate with each other using communication, comes to be hot issue [56] [62]. Many research mention that string stability is greatly improved by using communication information between vehicles and it is known that string stability can be satisfied more easily by feeding back advanced acceleration information using CAAC.

However, there are still problems such as communication error and its delay [63]. It is also big risk for the vehicles that they can lost their signals. Therefore we still need to establish communication-less relative distance observation and control.

7.2.2 Vision based ACC controller design

From the results of chapter 6, stereo vision sensor can estimate both velocity and depth. [64] is one of the researches to achieve ACC with a vision-aided system. Then, with the estimated information we can

derive the vision based ACC controller as:

$$u_i = F_1(\hat{d}_i - d_{ref,i}) + F_2\dot{\hat{d}}_i. \quad (7.5)$$

Eq. (7.5) means ACC controller can be written as state feedback controller.

From Fig. 7.2, the state-space plant model is converted to

$$\begin{aligned} \frac{d}{dt} \begin{pmatrix} p_i \\ v_i \\ a_i \end{pmatrix} &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau} \end{pmatrix} \begin{pmatrix} p_i \\ v_i \\ a_i \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ \frac{1}{\tau} \end{pmatrix} u_i. \\ &\equiv \left(\frac{d}{dt} \mathbf{x}_i = A_{cp} \mathbf{x}_i + B_{cp} u_i \right) \end{aligned} \quad (7.6)$$

So, substituting $d_i = p_{i-1} - p_i$ to Eq. (7.5) under constant inter-vehicle distance reference $d_{ref,i}$, we can rewrite Eq. (7.5) to

$$u_i = -F_1\hat{p}_i - F_2\hat{v}_i - F_1d_{ref,i} + (F_1\hat{p}_{i-1} + F_2\hat{v}_{i-1}). \quad (7.7)$$

Eq. (7.7) means state feedback gain of the ACC problem in Fig. 7.2 is

$$\begin{aligned} u_i &= \begin{pmatrix} -F_1 & -F_2 & 0 \end{pmatrix} \begin{pmatrix} \hat{p}_i \\ \hat{v}_i \\ \hat{a}_i \end{pmatrix} + \text{ref} + \text{disturbance}. \\ &\equiv (u_i = F_{cp}\hat{\mathbf{x}}_i + \text{ref} + \text{disturbance}) \end{aligned} \quad (7.8)$$

Substituting $\tau = 0.2$ and each parameters of Table. 7.1, controller poles in Fig. 7.3 are calculated with the eigenvalue of the $A_{cp} + B_{cp}F_{cp}$.

7.3 Feedback control simulation with state-feedback controller

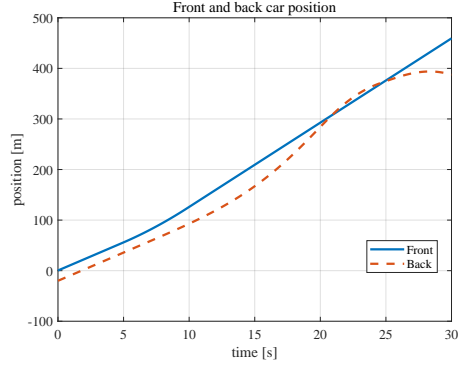
This section demonstrates the comparison of the EKF and pole-placement observer discussed in chapter 6.

With the stereo sensor fusion estimation results, we can extract camera depth Z and its time derivative \dot{Z} : that is equivalent to inter-vehicle distance d_i for the follower vehicle. Our simulation supposes two cars platooning model with vision-based state feedback control:

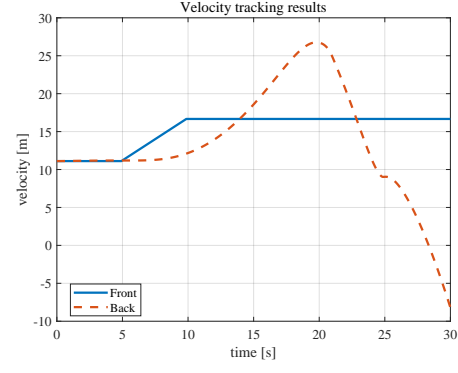
$$u_i[k+1] = \begin{pmatrix} F_1 & F_2 \end{pmatrix} \begin{pmatrix} \hat{d}_i[k] \\ \hat{\dot{d}}_i[k] \end{pmatrix} - F_1d_{ref} \quad (7.9)$$

while, the controller parameter F_1, F_2 in this simulation is chosen from the class C controller in Table. 7.1.

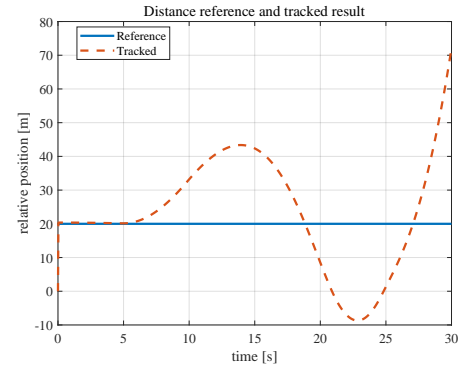
Sampling time and observer pole patterns in this simulation are the same as the chapter 6 and shown in Table. 6.1. The plant in Eq. (7.6), which imitates the car dynamics, is also discretized with the camera sampling 33 ms and $\tau = 0.2$, and it is controlled by the controller expressed at Eq. (7.9).



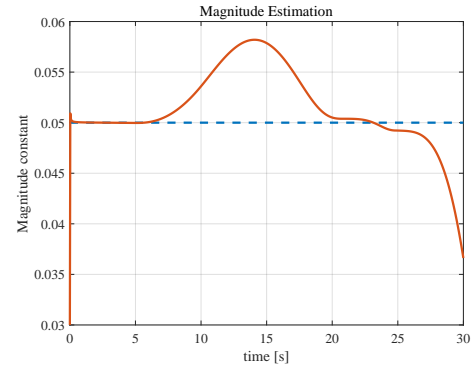
(a) Position tracking results.



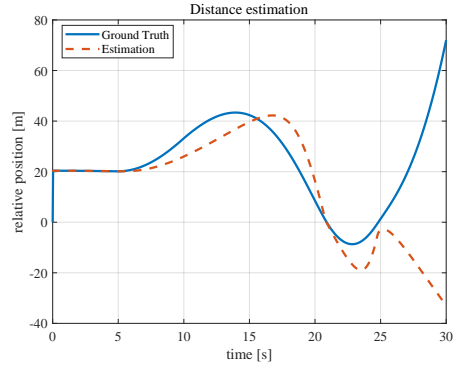
(b) Velocity tracking results.



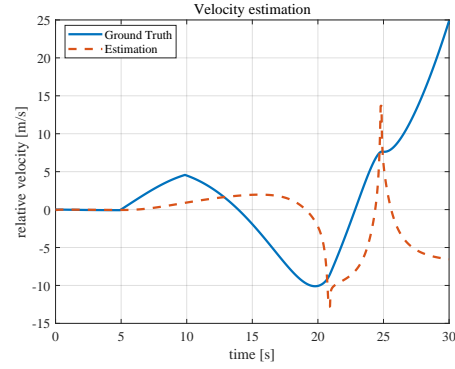
(c) Relative distance control results.



(d) Magnitude estimation results results.



(e) Relative position estimation results.



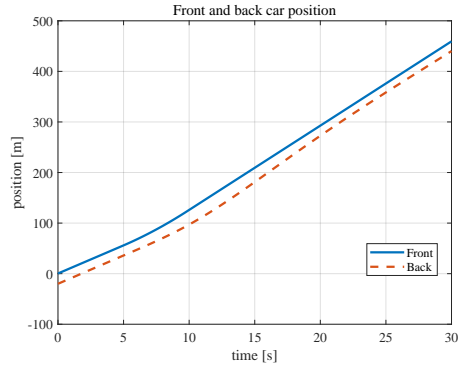
(f) Relative velocity estimation results.

Figure 7.4 EKF-estimation-based ACC results with the controller class C.

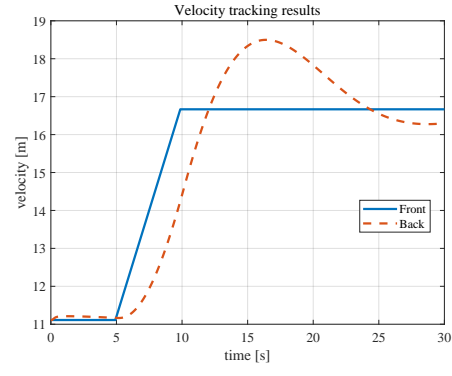
Fig. 7.4 and Fig. 7.5 shows the platooning systems response with EKF and pole-placement-observer-based sensor fusion method without sensor noises to evaluate the tracking response. The noise design in EKF used in Fig. 7.4 and the pole design of observer in Fig. 7.5 are the same with chapter 6, while the control results in Fig. 7.4 diverges.

Fig. 7.6 and Fig. 7.7 show the continuous poles of each observation filter and controller. The bad tracking reason shown in Fig. 7.4(c) can be considered as large estimation delay caused by slow observation filter poles.

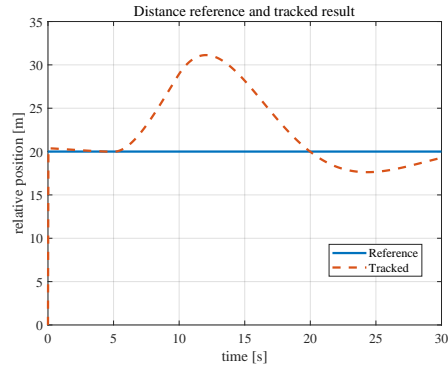
Then, the same tracking control simulations are held with the sensor noises in Table. 6.1. Despite the



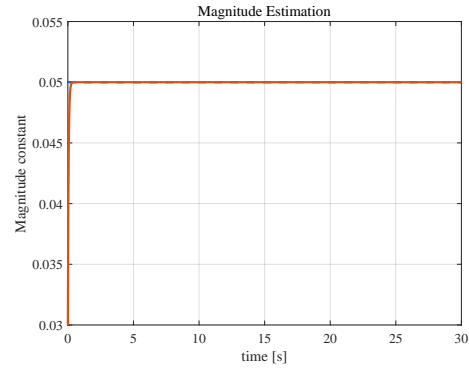
(a) Position tracking results.



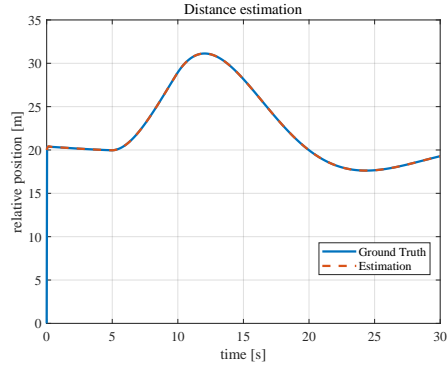
(b) Velocity tracking results.



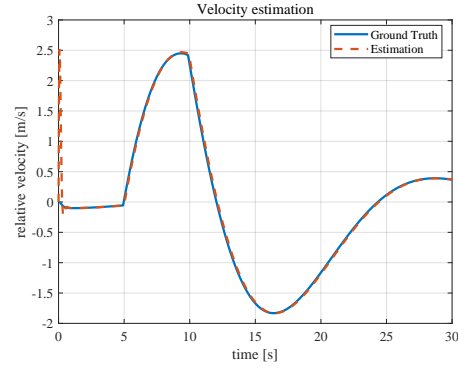
(c) Relative distance control results.



(d) Magnitude estimation results results.



(e) Relative position estimation results.



(f) Relative velocity estimation results.

Figure 7.5 Pole-placement-observer-estimation-based ACC results with the controller class C.

tracking results in Fig. 7.4, the EKF-based ACC at Fig. 7.8 successfully converged with a certain noise existence. Compared with the observer-based ACC at Fig. 7.9, EKF-based estimation can suppress noise influence drastically.

However, the inter-vehicle distance and velocity tracking comparison in Fig. 7.10 and Fig. 7.11 indicates that the observer-based ACC has a smaller overshoot and faster convergence.

Thus, these results mean that the "better" state observation will not lead to better control performance.

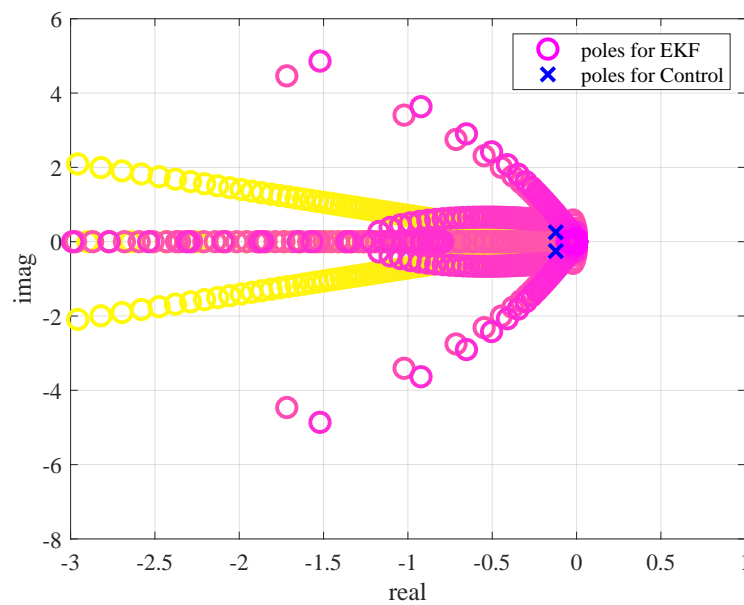


Figure 7.6 Relative position tracking results comparison with noise and the controller class C.

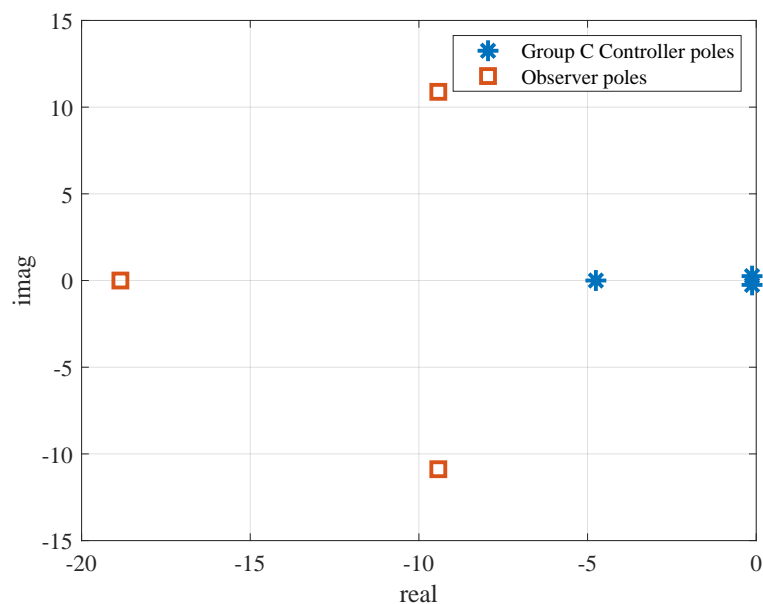


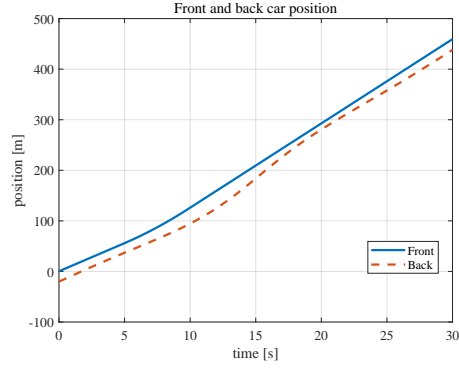
Figure 7.7 Velocity tracking results comparison with noise and the controller class C.

7.4 Conclusion

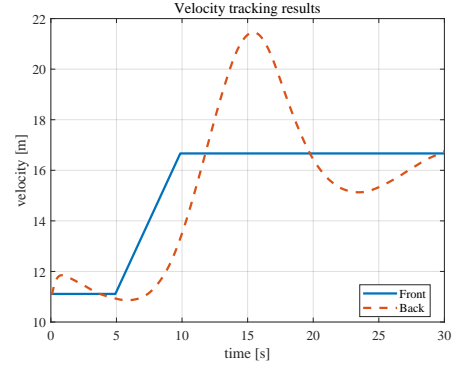
The ACC simulation results in this chapter indicates the importance of the co-design of the controller and observer.

In the sensor fusion issue, the probabilistic approaches such as EKF are popular, but they often need try-and-error covariance design. The observer-based approach is inferior in reducing the noise of the estimation but achieved better performance in the actual ACC simulation. This is because plant dynamics functioned as a low pass filter for the Gaussian sensor noise.

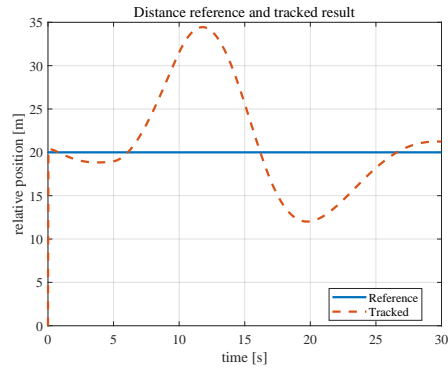
Concerning the co-design of sensing and control, it is much easier for the engineers to tune the pole assignment than noise covariance to design transient response. So, the proposed switching pole-assigned observer is effective in a sensor fusion and controller co-design problem.



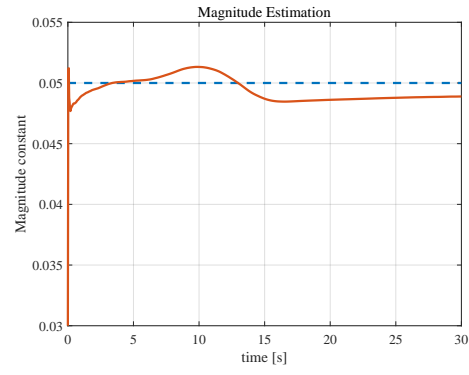
(a) Position tracking results.



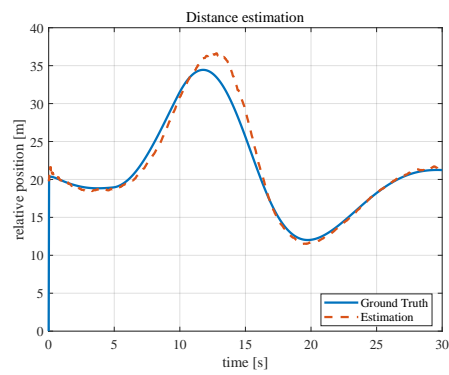
(b) Velocity tracking results.



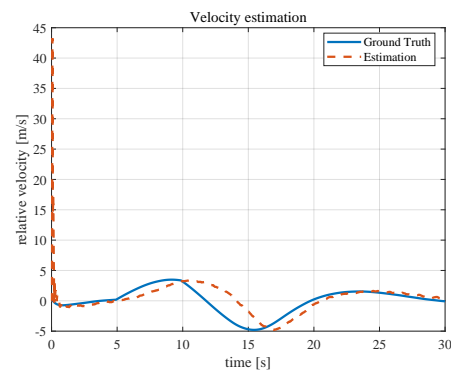
(c) Relative distance control results.



(d) Magnitude estimation results results.

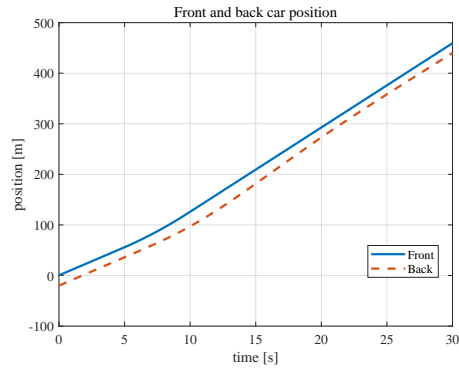


(e) Relative position estimation results.

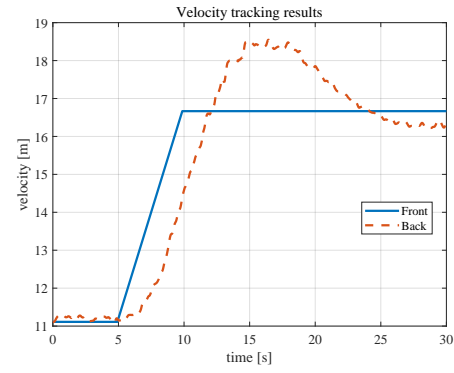


(f) Relative velocity estimation results.

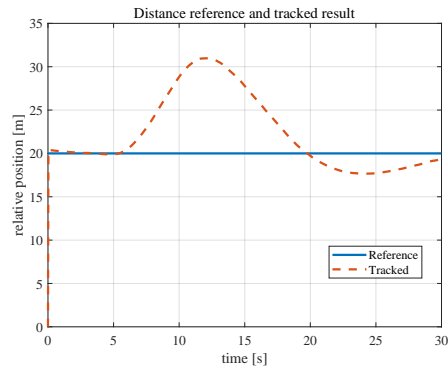
Figure 7.8 EKF-estimation-based ACC results with sensing noise and the controller class C.



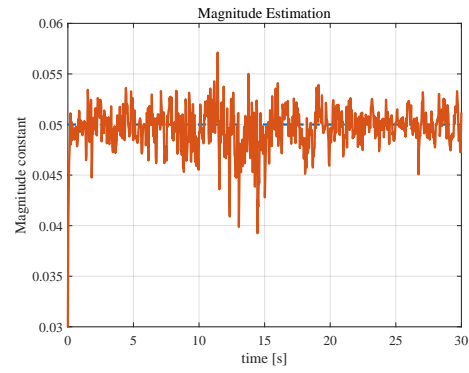
(a) Position tracking results.



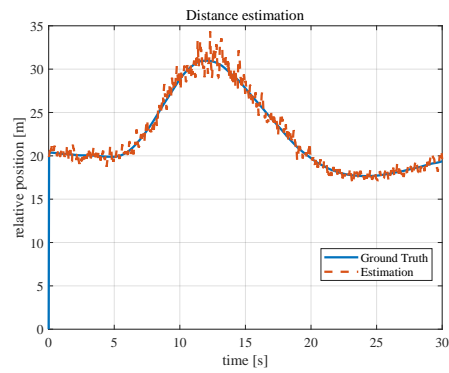
(b) Velocity tracking results.



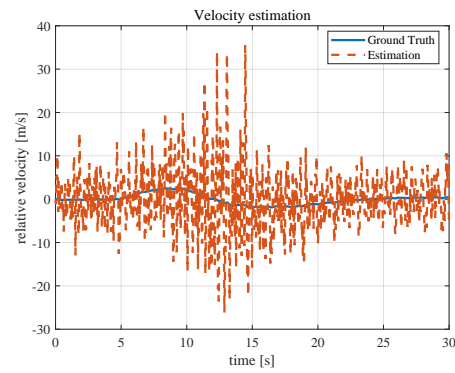
(c) Relative distance control results.



(d) Magnitude estimation results results.



(e) Relative position estimation results.



(f) Relative velocity estimation results.

Figure 7.9 Pole-placement-observer-estimation-based ACC results with sensing noise and the controller class C.

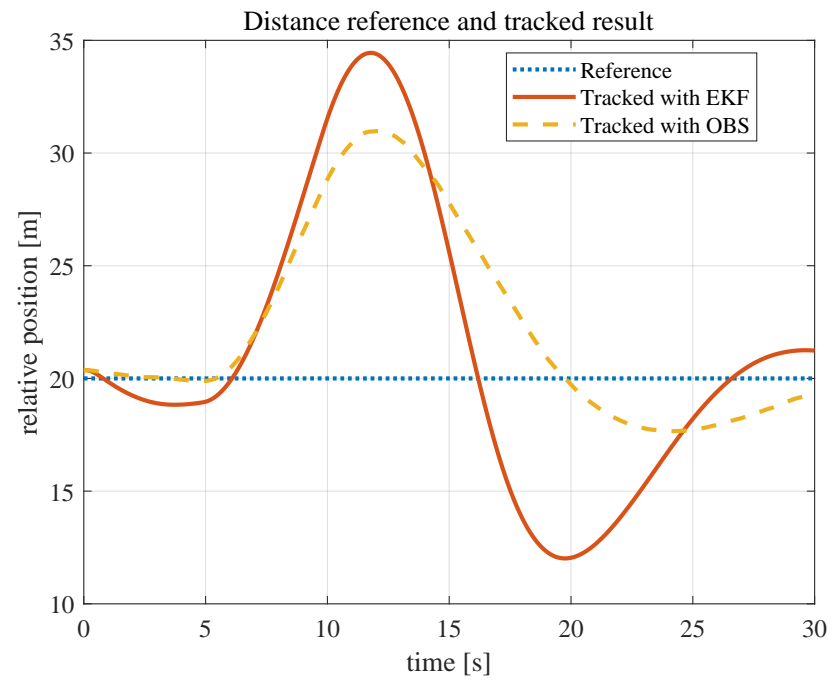


Figure 7.10 Relative position tracking results comparison with noise and the controller class C.

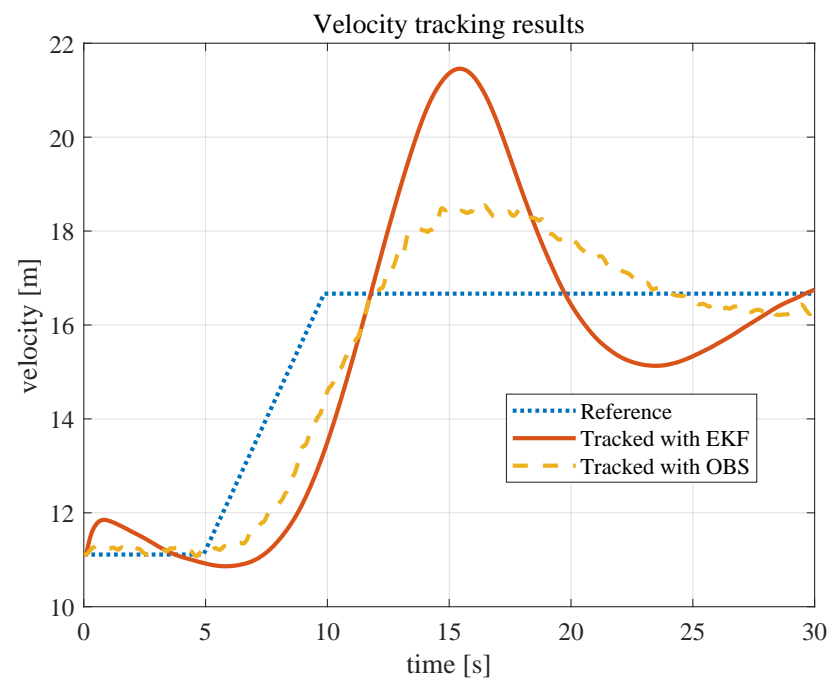


Figure 7.11 Velocity tracking results comparison with noise and the controller class C.

Chapter 8

Observer Designs in Terms of Gaussian Sensor Noise and Convergence Speeds

8.1 Motivation

In this section, we will discuss about general filter-type observer design technique considering both gaussian sensor noise and disturbance suppression in frequency domain. Filter design with modern control theory often uses eigen values to control such as its error convergence speed. On the other hand, state estimation using a noisy sensor often takes a stochastic approach such as a Kalman filter.

The purpose of this section is to clarify the effect on the state estimation of observation noise when a frequency-shaped gain is set in a filter-type discrete observer. This problem is highly related with sensor selection and controller co-design problem [65] [66].

8.1.1 Related work

According to Ono et al. of Hiroshima City University [67], conducted a numerical search for DOB designed by the Govinus method using the upper limit of the estimated error of disturbance expressed as the sum of impulse responses as an evaluation function. The optimal feedback gain is calculated.

There is also a paper that applies frequency-dependent weights to linear quadric gaussian (LQG) control [68]. As a technique, the weight function designed in the frequency domain is converted into real-time weights using Parseval's theorem. In other words, a filter is inserted before and after the original plant, and the problem of LQG is input and output It is equivalent to solving. In this case, only frequency shaping is performed, and there is a recognition that the same problem consciousness has shifted from the pursuit of design freedom to the H_2 problem and eventually the mixed optimization problem of H_2 and H_∞ norms.

As a related study, Kase et al. of the Osaka Institute of Technology have proposed a method to minimize the control input estimation error while ensuring the stability of the feedback system [69]. In this case, it is not H_2 optimal in the end, and it is possible to see that the optimization location is different because it is optimizing where it is easy to solve.

8.1.2 Proposed approach

There are two possible approaches. This is a pattern that puts the concept of poles in the KF-based method (noise-based design) and a pattern that puts noise evaluation items in the pole-based design.

The former is an adaptive observer, and the latter is in the same form as a normal observer. In this paper, the latter approach is used for the time being.

8.2 Fixed gain observer considering observation error

Consider a linear discrete state equation that explicitly considers the following observation noise: $\mathbf{x} \in \mathcal{R}^n$, $\mathbf{A} \in \mathcal{R}^{n \times n}$. In order to simplify the discussion, it is assumed that the system is discrete and the system is observable and controllable, and there is no system noise.

$$x_{k+1} = Ax_k + Bu_k \quad (8.1)$$

$$y_k = Cx_k + \omega_k \quad (8.2)$$

$\omega \in N(0, \Omega)$ in Eq. (8.2) represent Gaussian observation noise with the sensor.

8.2.1 Error and covariance study on discrete fixed gain observer

We set the filter type observer of the system in Eq. (8.2). Eq. (8.3) means prediction step and Eq. (8.4) means correction step.

$$\hat{x}_{k+1|k} = A\hat{x}_k + Bu_k \quad (8.3)$$

$$\hat{x}_{k+1} = \hat{x}_{k+1|k} + K(y_{k+1} - C\hat{x}_{k+1|k}) \quad (8.4)$$

Then, when we set the covariance of the estimated state \hat{x}_k as $V(\hat{x}_k) = \mathbf{P}_k$, recurrence relationships become

$$V(\hat{x}_{k+1|k}) = V(A\hat{x}_k) = AP_kA^\top \quad (8.5)$$

$$P_{k+1} = V((I - KC)\hat{x}_{k+1|k}) + V(Ky_{k+1}). \quad (8.6)$$

Merging Eq. (8.5) and Eq. (8.6) results in

$$P_{k+1} = (I - KC)P_{k+1|k}(I - KC)^\top + K\Omega K^\top \quad (8.7)$$

$$= (A - KCA)P_k(A - KCA)^\top + K\Omega K^\top \quad (8.8)$$

while $V(y_k) = \Omega$ was applied to simplify equations.

Thus, with an initial estimation covariance P_0 , the covariance at certain timing n can be written as:

$$P_n = (A - KCA)^n P_0 (A - KCA)^{n^\top} + \sum_{k=0}^{n-1} (A - KCA)^k K\Omega K^\top (A - KCA)^{k^\top} \quad (8.9)$$

, so with the perfect system identification covariance of estimation can be predicted from observer gain

and state-equations.

This covariance will converge only when the observer gain K satisfy the condition to stabilize following error system for estimation error $e_k = \mathbf{x}_k - \hat{\mathbf{x}}_k$:

$$e_{k+1} = (I - KC)Ae_k. \quad (8.10)$$

, which is equal to the condition that the every eigenvalue of $(I - KC)A$ fits in a unit circle.

With those requirement, the estimation covariance converge to $P_\infty = \lim_{n \rightarrow \infty} P_n$, which satisfies

$$P_\infty = \sum_{n=0}^{\infty} (A - KCA)^n K \Omega K^\top (A - KCA)^{n\top}. \quad (8.11)$$

P_∞ in Eq. (8.11) is given as a a solution of the discrete Lyapunov equation for P in following equations:

$$RPR^\top - P + Q = 0 \quad (8.12)$$

$$R = A - KCA, \quad Q = K\Omega K^\top \quad (8.13)$$

Thus, observer estimation error can be evaluated via converged covariance which can be calculated in advance. This means observer design can control not only convergence speed via pole placement but sensor noise influence.

It is also indicated that we can vary the observation noise Ω and sampling time by changing the sensor, which shows sensing algorithm choice is also able to belong to the observer design.

Minimizing trace of this Lyapunov equation $\text{trace}(P_\infty)$ can be converted into H_2 norm optimization [70] [71].

8.2.2 Simulative varidation for steady-state covariance estimation

For checking the correctness of the Eq. (8.11), this section shows the simulation.

Assume continuous system,

$$\begin{aligned} A_c &= \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -2 & -3 & -4 \end{pmatrix}, \quad B_c = \begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \\ C &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \end{aligned} \quad (8.14)$$

with sensor error covariance

$$\Sigma = \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad (\sigma_1, \sigma_2) = (0.1, 0.2). \quad (8.15)$$

Sampling time for discretization was set to 0.1ms and discrete poles were placed on $(-1, -1.5, -2)$.

Pre-calculation results of Eq. (8.11) was shown as $P_{calculated}$ in Eq. (8.16) and actual estimation value covariance is shown as P_{actual} . In the actual calculation, 300 times simulation with same initial estimation value and different seeds-oriented Gaussian noise was added to observation, then covariances were evaluated at the same timing; 10s after the simulation.

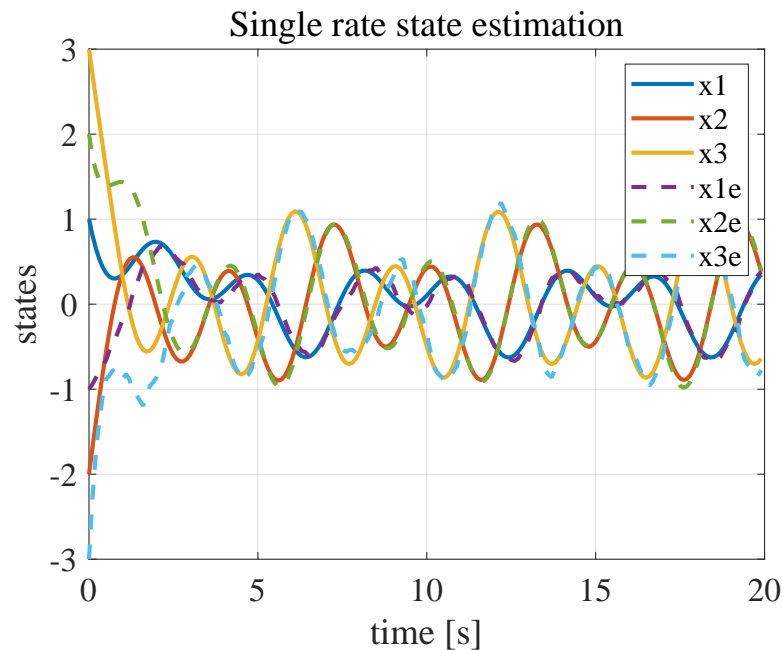


Figure 8.1 State observation results.

$$\begin{aligned}
 P_{actual} &= \begin{pmatrix} 0.0026 & -0.0014 & -0.0026 \\ -0.0014 & 0.0020 & -0.0000 \\ -0.0026 & -0.0000 & 0.0048 \end{pmatrix} \\
 P_{calculated} &= \begin{pmatrix} 0.0027 & -0.0015 & -0.0027 \\ -0.0015 & 0.0021 & -0.0000 \\ -0.0027 & -0.0000 & 0.0049 \end{pmatrix}
 \end{aligned} \tag{8.16}$$

Eq. (8.16) shows the correctness of the explained error estimation.

8.2.3 Study on sensor choice problem

In the former simulation, two observations are considered, but we can reduce one observation with reduced sensing matrix:

$$C_{reduced} = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix}. \tag{8.17}$$

In this sensor-reduced system, the diagonal part of the finite covariance matrix become (0.0005, 0.0013, 0.0011), which is smaller than original sensing result (0.0027, 0.0021, 0.0049). This results means that the additional sensor is too noisy to interrupt other sensor estimation under the observer design and the sensor noise in previous section.

So, in terms of noise propagation, adding redundant sensor is bad choice rather than improving performance.

8.3 Sensor noise effect evaluation with state feedback and observations

Consider an observable and controllable LTI system to evaluate how the sensor noise affect to the estimation and control results. Assuming system in Fig. 8.2 with gaussian system and sensor noise, we can write down its discrete time state equation as:

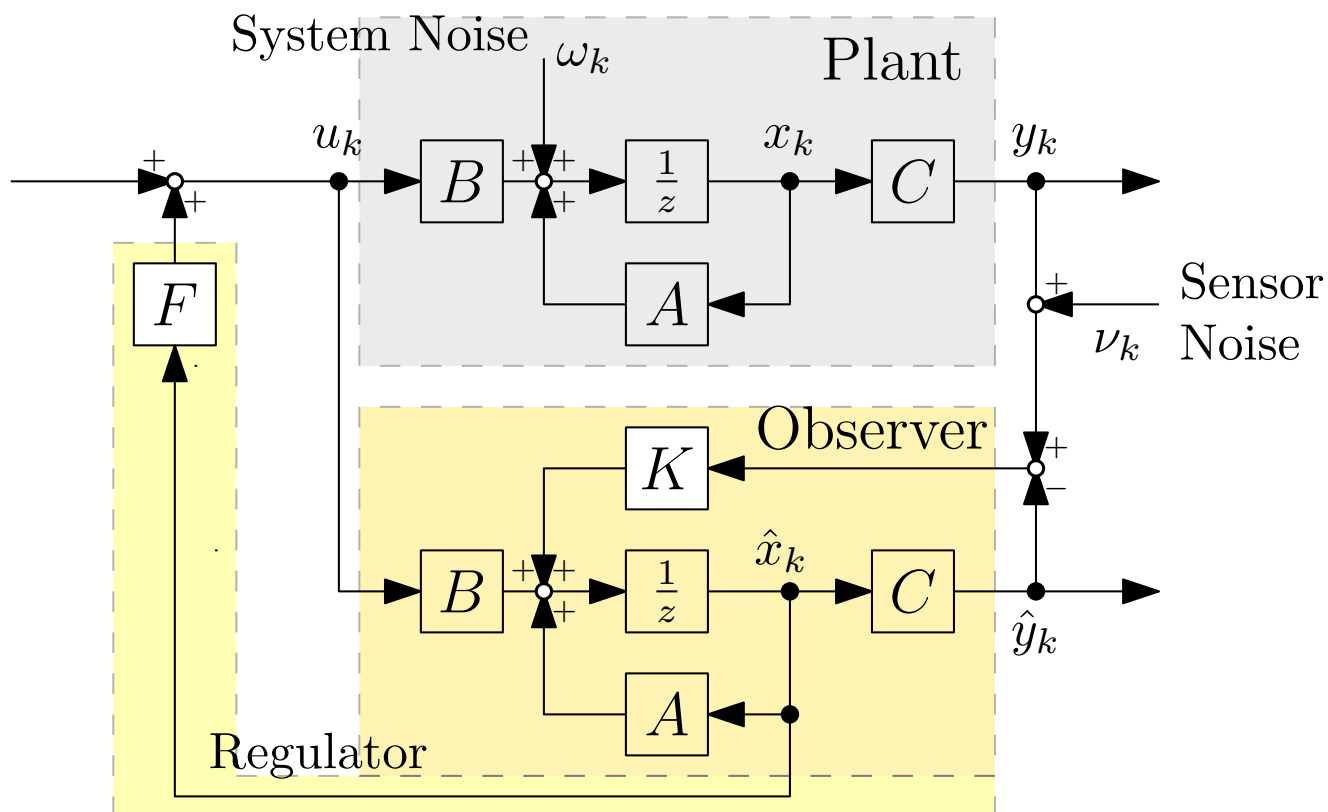


Figure 8.2 Discrete state feedback and observation system considering sensor and system noise.

$$x_{k+1} = Ax_k + Bu_k + \omega_k \quad (8.18)$$

$$y_k = Cx_k + \nu_k. \quad (8.19)$$

Then, the filter type observer, which uses $t = k + 1$ sensor value to estimate $t = k + 1$ value, can be written with appropriate feedback gain K .

$$\hat{x}_{k+1|k} = A\hat{x}_k + Bu_k \quad (8.20)$$

$$\hat{x}_{k+1|k+1} = \hat{x}_{k+1|k} + K(y_{k+1} - C\hat{x}_{k+1|k}) \quad (8.21)$$

The upper equation can be combined into Eq. (8.22).

$$\hat{x}_{k+1} = A\hat{x}_k + Bu_k + K(y_{k+1} - C(A\hat{x}_k + Bu_k)) \quad (8.22)$$

Also, we define the state feedback input as

$$u_k = F\hat{x}_k + Gv_k \quad (8.23)$$

and estimation error as

$$e_k = \pm(\hat{x}_k - x_k) \quad (8.24)$$

Both plus-and-minus definition in Eq. (8.25) can be possible.

The error equation in Eq. (8.25) is derived by applying Eq. (8.24) to the results of subtraction of Eq. (8.22) from Eq. (8.18).

$$e_{k+1} = (A - KCA)e_k + \underline{(I - KC)\omega_k - K\nu_k} \quad (8.25)$$

while, underline means the error distributed from incoming noises.

Similarly, Eq. (8.26) is obtained by substituting Eq. (8.23) and Eq. (8.24) for Eq. (8.18).

$$x_{k+1} = (A + BF)x_k \pm BFe_k + BGv_k + \underline{\omega_k} \quad (8.26)$$

The following equation is obtained by combining Eq. (8.25) and Eq. (8.26) to form a system of expanding the state and observer errors.

$$\begin{aligned} \begin{pmatrix} x_{k+1} \\ e_{k+1} \end{pmatrix} &= \begin{pmatrix} A + BF & \pm BF \\ 0 & A - KCA \end{pmatrix} \begin{pmatrix} x_k \\ e_k \end{pmatrix} + \begin{pmatrix} BG \\ 0 \end{pmatrix} v_k \\ &\quad + \underline{\begin{pmatrix} I \\ I - KC \end{pmatrix} \omega_k + \begin{pmatrix} 0 \\ -K \end{pmatrix} \nu_k} \end{aligned} \quad (8.27)$$

Since the state transition matrix $A_{EX} = \begin{pmatrix} A + BF & \pm BF \\ 0 & A - KCA \end{pmatrix}$ in the extended system is a block triangular matrix, if the pair is a stable observer and controller, the poles for the convergence can be individually designed.

8.3.1 Observation noise effect to the estimation and controlled error

In this section, we discuss how observation noise ν_k affect the system assuming that there is no system noise $\omega_k = 0$.

This also assumes that the observation noise is a Gaussian error with a mean of 0 and a variance of σ_ν^2 . When the expansion system of Eq. (8.27) is simplified as Eq. (8.28),

$$X_{E,k+1} = A_{EX}X_{E,k} + G_{EX}v_k + \underline{K_{EX}\nu_k} \quad (8.28)$$

$$\text{while, } G_{EX} = \begin{pmatrix} BG \\ 0 \end{pmatrix}, K_{EX} = \begin{pmatrix} 0 \\ -K \end{pmatrix} \quad (8.29)$$

The terminal value of the error covariance $V(X_{E\infty})$ in the system $X_{E,k}$, which is driven by observation noise ν_k , is represented by an infinite series in Eq. (8.30).

$$\begin{aligned} V(X_{E\infty}) &= A_{EX}^\infty V(X_{E0}) A_{EX}^{\infty\top} + \\ &\quad \sum_{k=0}^{\infty} A_{EX}^k K_{EX} \sigma_\nu^2 K_{EX}^\top A_{EX}^{k\top} \end{aligned} \quad (8.30)$$

This terminal value is equivalent to P satisfying the Lyapunov equation in Eq. (8.32).

$$RPR^\top - P + Q = 0 \quad (8.31)$$

$$R = A_{EX}, Q = K_{EX} \sigma_\nu^2 K_{EX}^\top \quad (8.32)$$

The diagonal components of the terminal value $V(X_{E\infty})$ of the resulting covariance matrix become the value of the variance of each state vector in the expanded system. It can be used to predict these at the stage of controller and observer design.

8.3.2 Observation and control separation in noise propagation

In the eigenvalue design of the extended system, it was shown that the controller and the observer can be designed separately from the features of the block triangular matrix.

In this section, we consider whether this separation holds in the noise covariance.

Since the matrix A_{EX} is a block triangular matrix, A_{EX}^k is also a block triangular matrix like:

$$A_{EX}^k = \begin{pmatrix} (A + BF)^k & \pm BF \sum_{i=0}^k (A + BF)^i (A - KCA)^{k-i-1} \\ 0 & (A - KCA)^k \end{pmatrix} \quad (8.33)$$

Each component of the series in the infinite series of Eq. (8.30) can be represented as

$$A_{EX}^k K_{EX} \sigma_\nu^2 K_{EX}^\top A_{EX}^{k\top} = \sigma_\nu^2 \begin{pmatrix} \mathcal{X}_k \mathcal{X}_k^\top & \mathcal{X}_k \mathcal{Y}_k^\top \\ \mathcal{Y}_k \mathcal{X}_k^\top & \mathcal{Y}_k \mathcal{Y}_k^\top \end{pmatrix} \quad (8.34)$$

$$\begin{aligned} \mathcal{X}_k &= BF \sum_{i=0}^k (A + BF)^i (A - KCA)^{k-i-1} K \\ \mathcal{Y}_k &= (A - KCA)^k K. \end{aligned} \quad (8.35)$$

\mathcal{Y}_k contains observer gain K , and \mathcal{X}_k contains controller gain F and observer gain K .

From this, the covariance of the control result and the observation result obtained by adding this element infinitely is the same for $\sum_k^\infty \mathcal{X}_k$ and $\sum_k^\infty \mathcal{Y}_k$. In addition, only the observer gain K must be considered for the covariance of the observation result, and both K and F must be considered when considering the covariance of the control result.

Whether the optimal observer and the optimal controller can be designed independently from the viewpoint of the Gaussian sensor noise depends on the proposition summarized from following hypothesis:

Hypothesis 1. *The optimal observer gain K which minimizes the final estimation error covariance $P_{obs} = \sum_k^\infty \mathcal{Y}_k$ also minimizes the control error covariance $P_{state} = \sum_k^\infty \mathcal{X}_k$.*

The proposal to solve this optimal gain with using matrix inequality is shown in appendix.

8.3.3 Practical observer design example with the linear stage parameters

This part shows the approximated solution to achieve optimal noise reduction and convergence rate under a certain conditions.

Suppose to control the linear stage used in the 6; the second-order LTI plant from input torque to encoder position output can be expressed as:

$$P_{linstage}(s) = \frac{1}{Ms^2 + Ds + K}. \quad (8.36)$$

Fig. 8.3 shows bode plots of the system identification results with chorp signal, and Table. 8.1 shows its variables.

With the parameters in Table. 8.1, a state-space expression containing the position and velocity as its

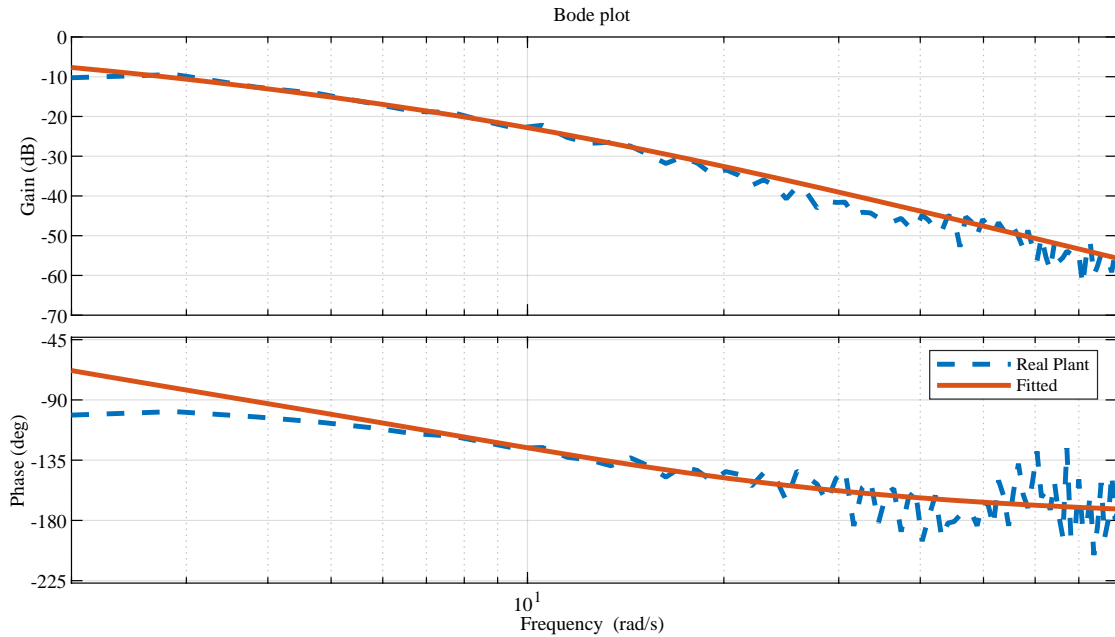


Figure 8.3 Bode Plots for the linear Stage.

Table 8.1 The linear stage parameter

Mass M	0.0936
Damping D	1.1226
Spring constant K	1.2783

state value is become Eq. (8.38).

$$A_c = \begin{pmatrix} 0 & 1 \\ -K/M & -D/M \end{pmatrix}, B_c = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \quad (8.37)$$

$$C_c = (1 \ 0), D_c = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (8.38)$$

We use the discretized plant in Eq. (8.40) with every 30 ms sampling of normal camera.

$$A_d = \begin{pmatrix} 0.9945 & 0.0251 \\ -0.3434 & 0.6930 \end{pmatrix}, B_d = \begin{pmatrix} 0.0004002 \\ 0.02514 \end{pmatrix}, \quad (8.39)$$

$$C_d = (1 \ 0), D_d = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (8.40)$$

In this simulation, we evaluated the gain of the observer designed with one parameter by comparing the steady-state covariance. The covariances are both calculated from the Lyapunov equation and direct calculation from 400 times trials.

The state feedback controller as a regulator is designed to have 5 rad poles of butterworth or overlaid-poles patterns. Then the observers were also designed to have butterworth or overlaid-poles patterns with different convergence rates. Fig. 8.4 and Fig. 8.8 shows the arranged poles of the observers and controller.

Fig. 8.5 and Fig. 8.9 indicate different trends in the trace of the observer and state covariance; the pole patterns which minimize observer error are not decreasing the state covariance well. Taking a look

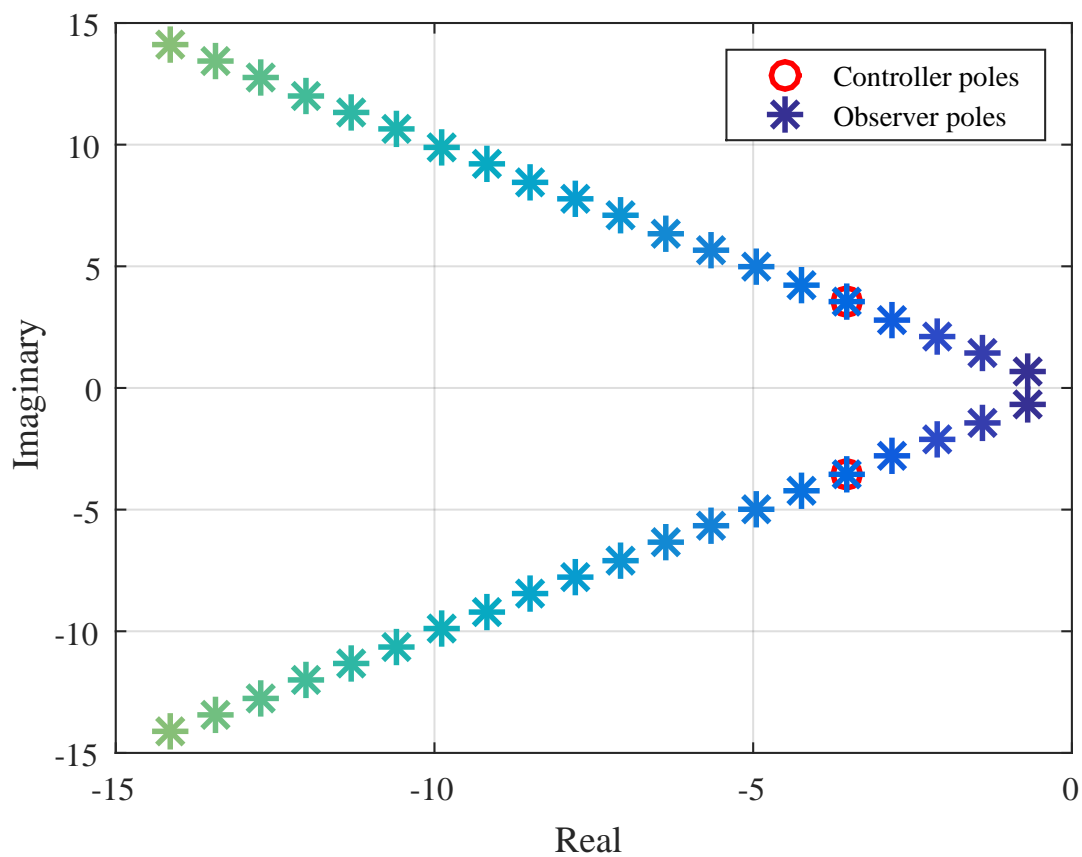


Figure 8.4 Butterworth poles for control and observation.

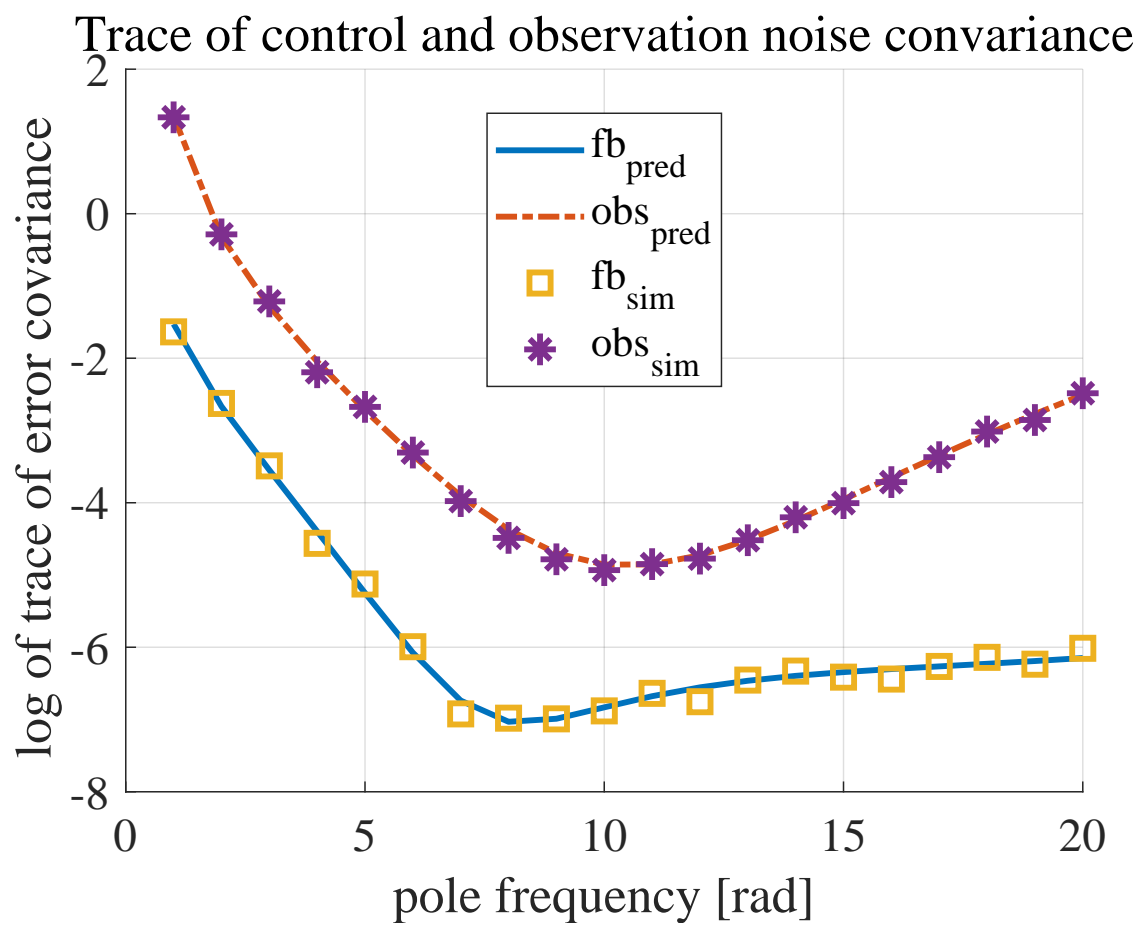


Figure 8.5 Observer noise evaluation via $trace(P_{obs})$ and state noise evaluation via $trace(P_{obs})$ with the butterworth pattern.

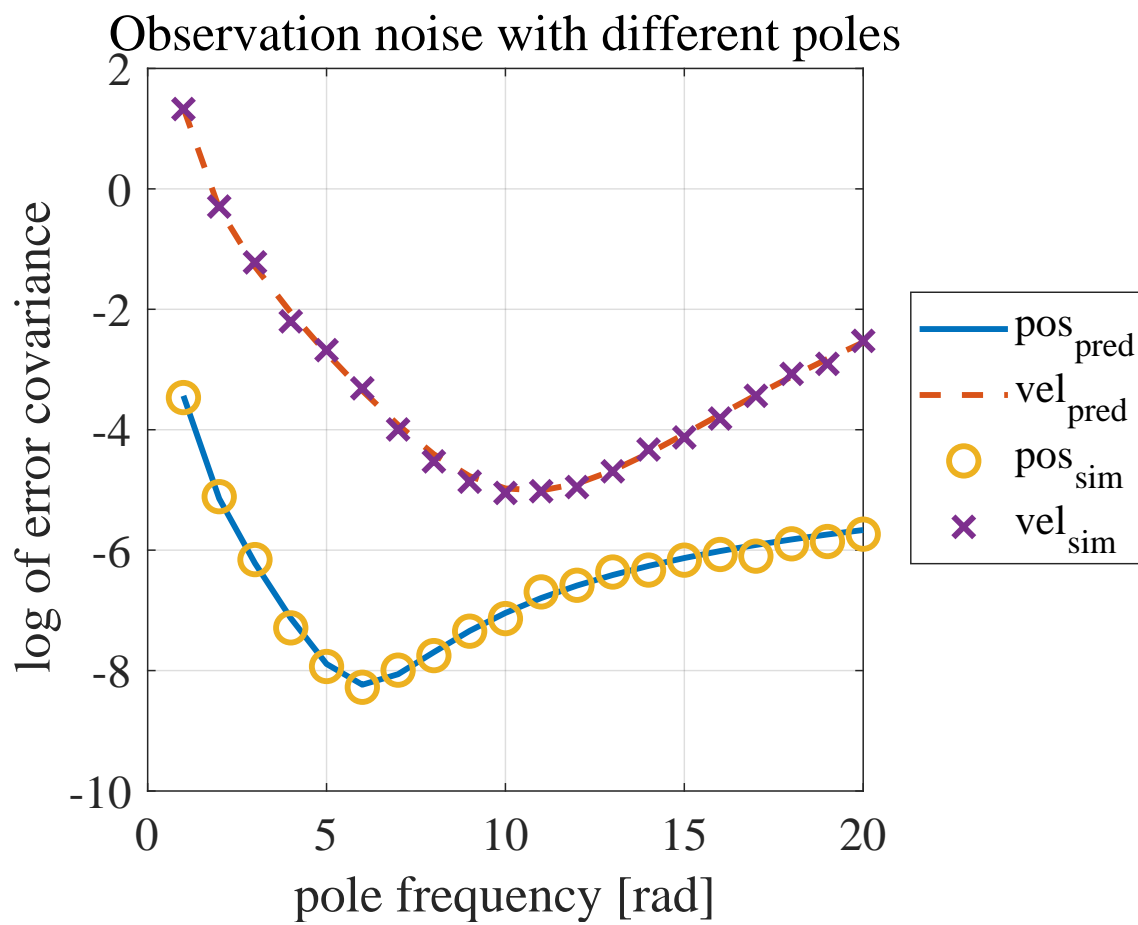


Figure 8.6 Observer noise prediction for each butterworth observation poles and simulation results.

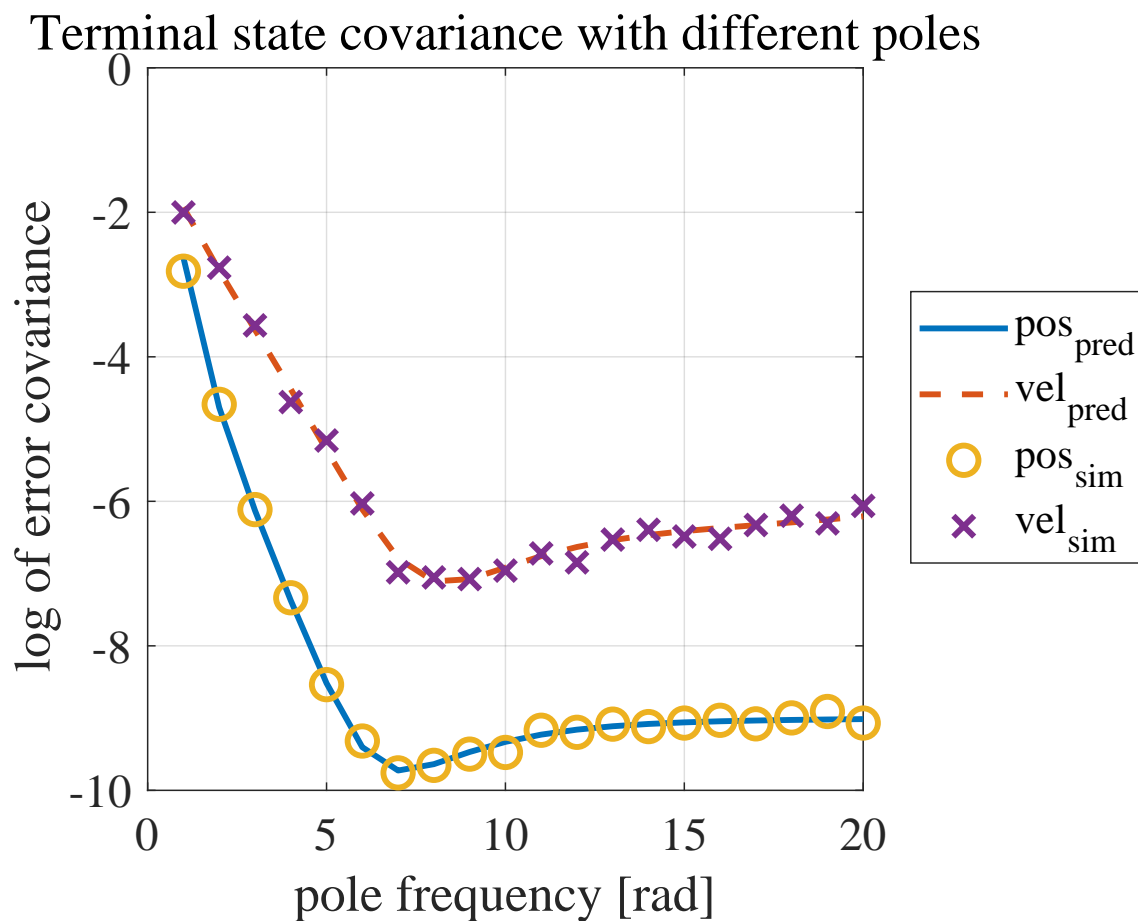


Figure 8.7 State noise prediction for each butterworth observation poles and simulation results

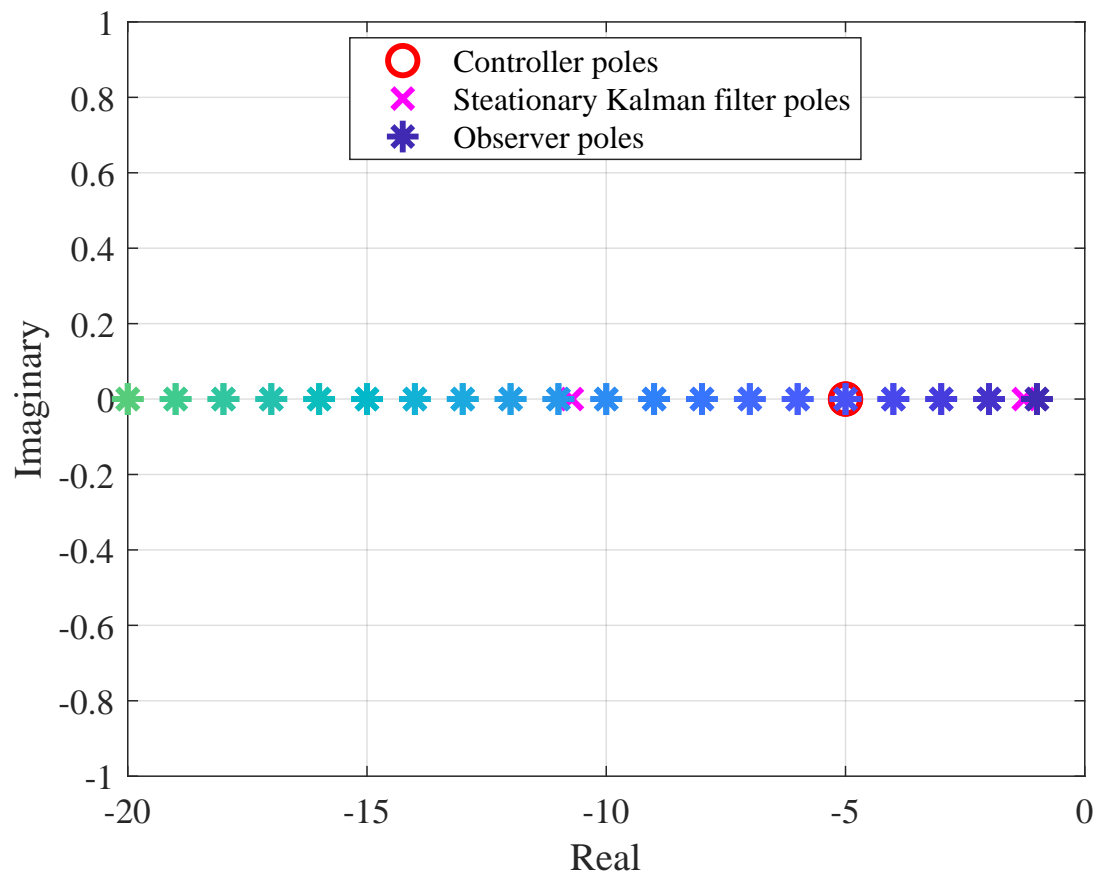


Figure 8.8 Overlapping poles for control and observation.

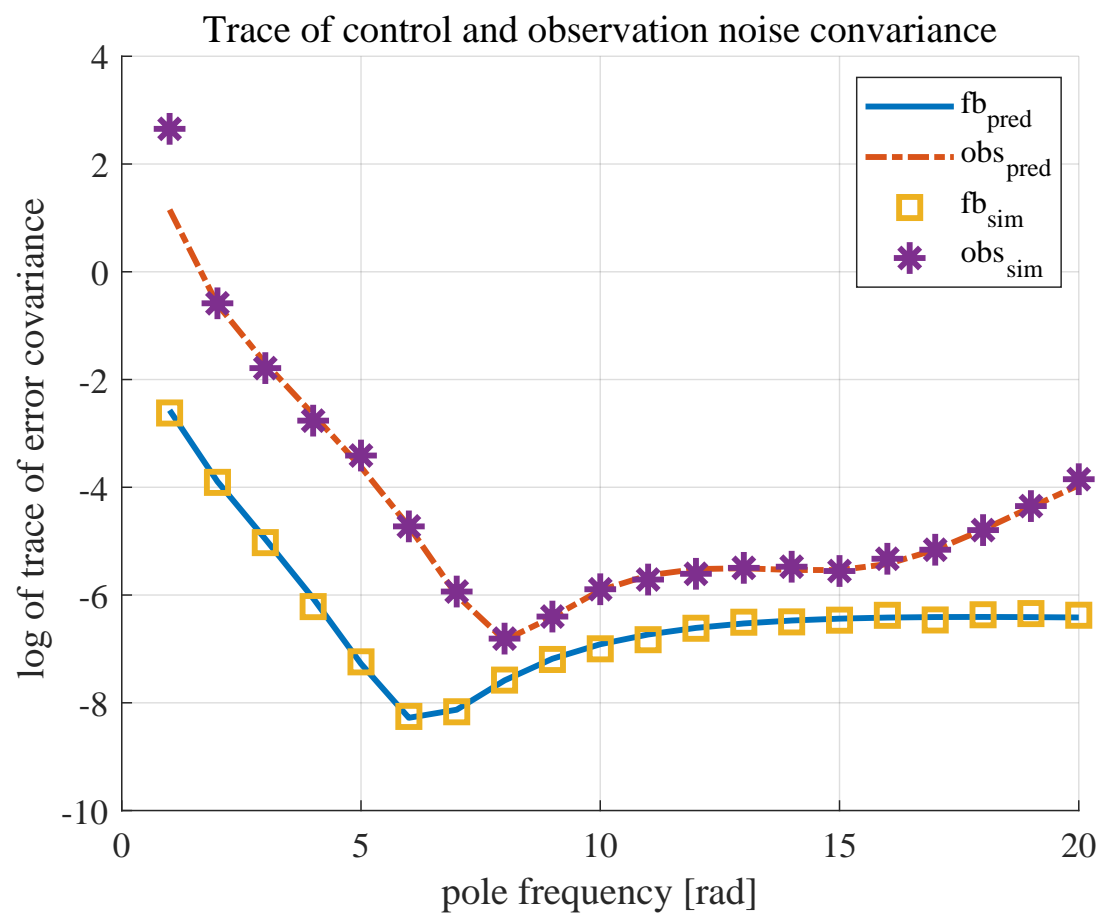


Figure 8.9 Observer noise evaluation via $\text{trace}(P_{obs})$ and state noise evaluation via $\text{trace}(P_{obs})$ with the overlapping pole pattern.

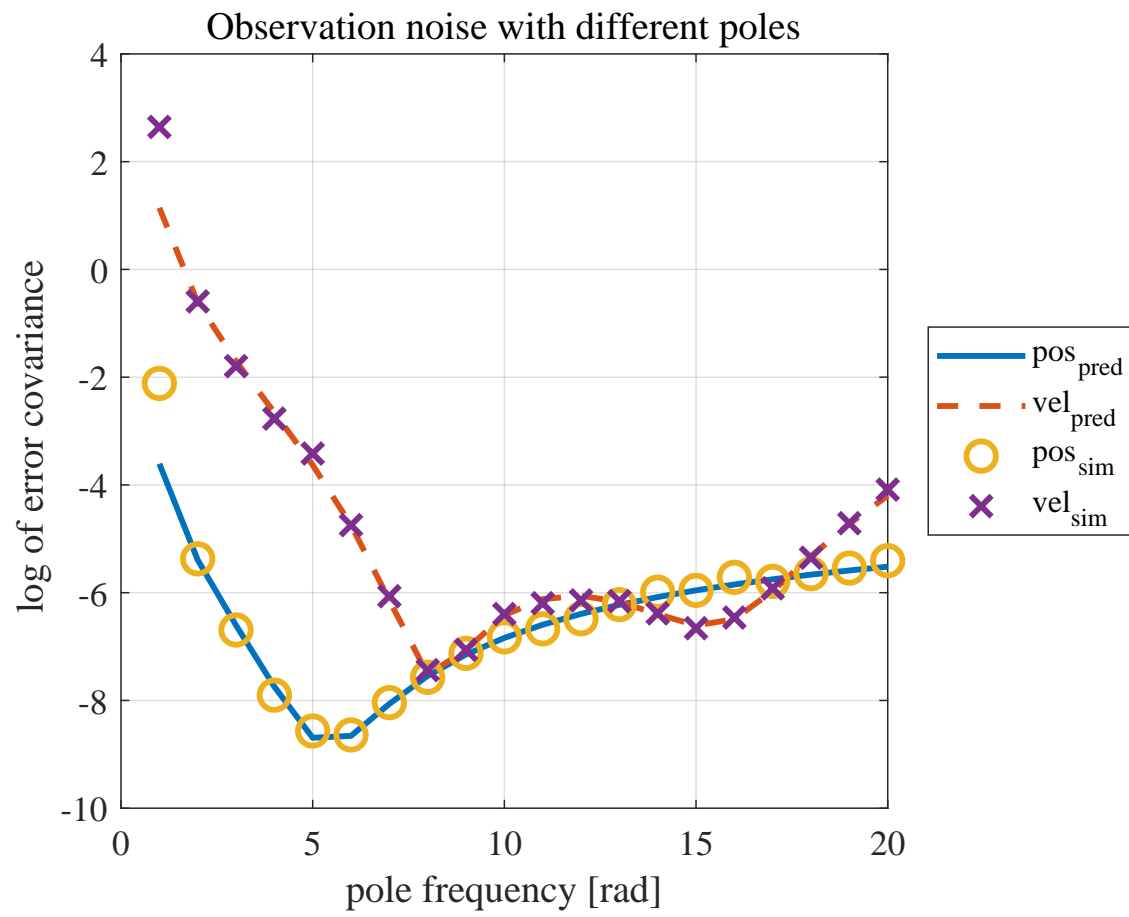


Figure 8.10 Observer noise prediction for each overlapping pole-based observation poles and simulation results.

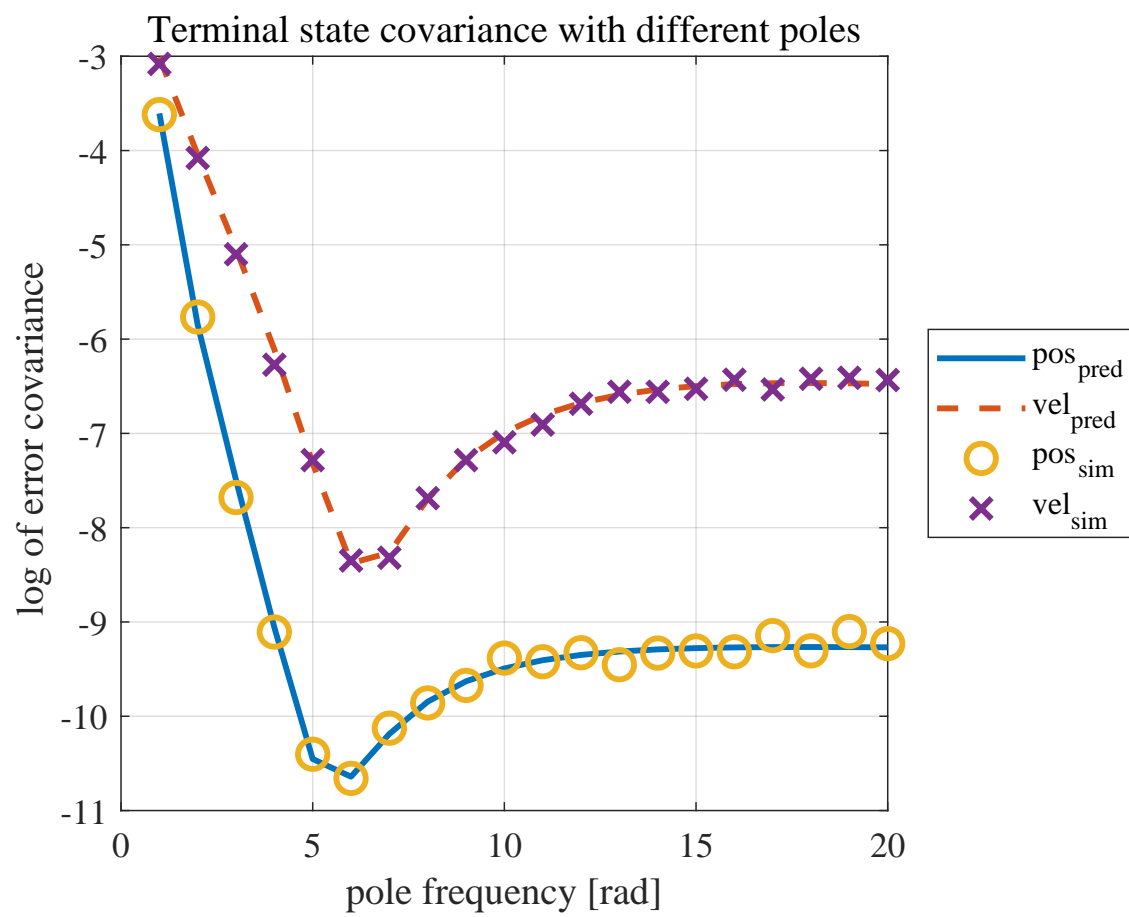


Figure 8.11 State noise prediction for each overlapping pole-based observation poles and simulation results

Table 8.2 Error analysis of run-time performance of our system on the KITTI (1241 x 376 px, 0.46 MP) dataset [7] in [8].

Method	Accuracy (%)			
	2px<	3px<	4px<	5px<
SGBM [73]	89	93.9	95.6	96.5
ELAS [74]	92.7	96.1	97.3	97.9
Line-Sweep [75]	72.6	81.2	84.7	86.7
Pillai2016 [8]	83.1	89.9	92.9	94.7

Table 8.3 Sensing noise assumption to calculate each matching method's variance and run-time from [8].

Method	Error distribution (%)				Variance		Run-time
	= 1px	= 3px	= 4px	= 5px	[pix]	[m]	[ms]
SGBM [73]	89	4.9	1.7	0.9	1.8943	2.2634	351.9
ELAS [74]	92.7	3.4	1.2	0.6	1.6088	1.9050	160.9
Line-Sweep [75]	72.6	8.6	3.5	2	2.9527	3.6461	70
Pillai2016 [8]	83.1	6.8	3	1.8	2.5058	3.0518	10.8

into each state variables shown in Fig. 8.6, Fig. 8.7, Fig. 8.10, and Fig. 8.11 it is also obvious that better observation does not necessarily lead better control performance.

Therefore, under limited conditions, observation and control cannot be said to be designed independently, and the noise evaluation must be designed according to the final control target.

8.3.4 Sensor choice considering effects of noise and its application for image processing algorithm evaluation

The last section demonstrated the noise-and-convergence-rates-based observation filter design. The proposed steady-state variance evaluation is also applicable to the sensor choice problem.

It is well known that image perception and its processing has speed-accuracy tradeoff(SAT) [58]. This section focuses on the algorithms used in a stereo disparity matching process [72] [8].

Table. 8.2 shows the accuracy of the each famous stereo matching methods introduced in [8].

From Table. 8.2, we create the simulated error distribution of each stereo matching methods. In our assumption, absolute disparity error distribution is estimated from Table. 8.2 and shown in Table. 8.3. Then, disparity variance and depth variance are calculated; we suppose ACC with 20m inter vehicle distance and the camera setting in KITTI dataset [7] $f = 645.24$, $b = 0.54$ m.

Since the four method shown in Table. 8.3 have different run-time according to [8], this section evaluates these methods with different sampling discrete models.

These processing methods are evaluated using the ACC controller and pole assignment estimator used in Chapter 7. We evaluate the following discretized plant obtained by discretizing Eq. (7.6) with sampling time T_s .

Table 8.4 Sensing method comparison result.

Method	T_s	σ^2	Variance with group A		Variance with group B		Variance with group C	
	[ms]	[m ²]	Distance [m]	Velocity [m/s]	Distance [m]	Velocity [m/s]	Distance [m]	Velocity [m/s]
SGBM [73]	351.9	2.2634	0.2066	0.2183	0.1752	0.1184	0.2609	0.1017
ELAS [74]	160.9	1.9050	0.0764	0.1155	0.0649	0.0624	0.0941	0.0496
Line-Sweep [75]	70	3.6461	0.0624	0.1029	0.0532	0.0557	0.0767	0.0439
Pillai2016 [8]	10.8	3.0518	0.0812	0.1296	0.0691	0.0701	0.0998	0.0553

$$\mathbf{x}_i[k+1] = A_{cp}\mathbf{x}_i[k] + B_{cp}u_i[k] \quad (8.41)$$

$$\mathbf{x}_i[k] = \begin{pmatrix} p_i[k] \\ v_i[k] \\ a_i[k] \end{pmatrix} \quad (8.42)$$

$$A_{dp} = \begin{pmatrix} 1 & T_s & \tau T_s - \tau^2 + \tau^2 e^{-\frac{T_s}{\tau}} \\ 0 & 1 & \tau - \tau e^{-\frac{T_s}{\tau}} \\ 0 & 0 & e^{-\frac{T_s}{\tau}} \end{pmatrix} \quad (8.43)$$

$$B_{dp} = \begin{pmatrix} \tau^2 - \tau T_s + \frac{T_s^2}{2} - \tau^2 e^{-\frac{T_s}{\tau}} \\ T_s - \tau + \tau e^{-\frac{T_s}{\tau}} \\ 1 - e^{-\frac{T_s}{\tau}} \end{pmatrix} \quad (8.44)$$

The observer poles were set to the 6π rad butteworth poles and state-feedback controller gain is set as $F_{dp} = \begin{pmatrix} -F_1 & -F_2 & 0 \end{pmatrix}$. The variance σ^2 and the sampling T_s were chosen from depth variance and run-time in Table. 8.3.

Finally, the steady-state variance of each state is calculated from Eq. (8.32) with the model in Eq. (8.42).

Table. 8.4 shows a comparison of control performance for each controller parameter shown in Table. 7.1, and variance of each state variable is illustrated in Fig. 8.12, Fig. 8.13, and Fig. 8.14. According to Table. 8.4, distance estimation using Line-Sweep [75] can minimize the relative position tracking variance under given assumptions and conditions of ACC.

Fig. 8.12 shows that ELAS [74] is superior in estimating distance and velocity because it has a smaller estimation error. As for the acceleration estimation, where the effect of the second derivative is significant, Pillari2016 [8] is superior since it has a fast run-time. However, as a result of the control, Line-Sweep [75] comes to achieve the best performance to suppress the dispersion of the relative distance. This can be considered as the processing noise was reduced by the feedback controller with slow convergence.

Also, comparing the controller groups in Table. 8.4 shows that group B can reduce the variance of the relative distance, so the analysis shown in this chapter should be performed for the observer, controller, and sensor parameters. Thus, it is possible to design a system that maximizes control performance comprehensively.

Furthermore, a comparison of the controller groups in Table. 8.4 shows that group B can reduce the variance of the relative distance. Thus, the proposed analysis in this chapter enables us to co-design the sensing, control, and observer from both noise and convergence speed.

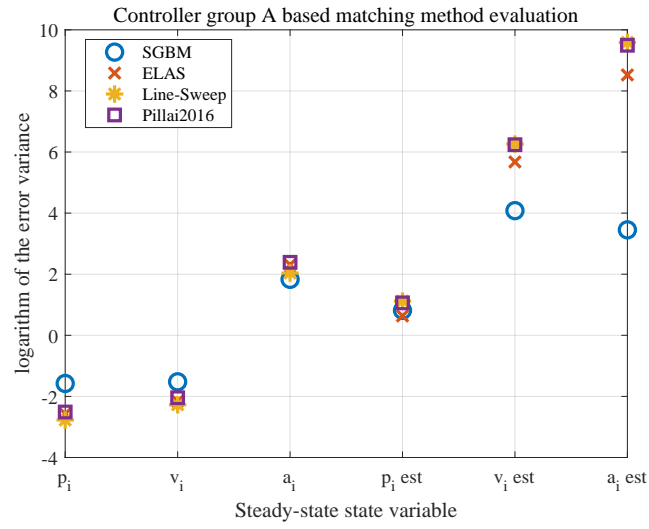


Figure 8.12 Log scale steady-state variance of each state variable with controller group A.

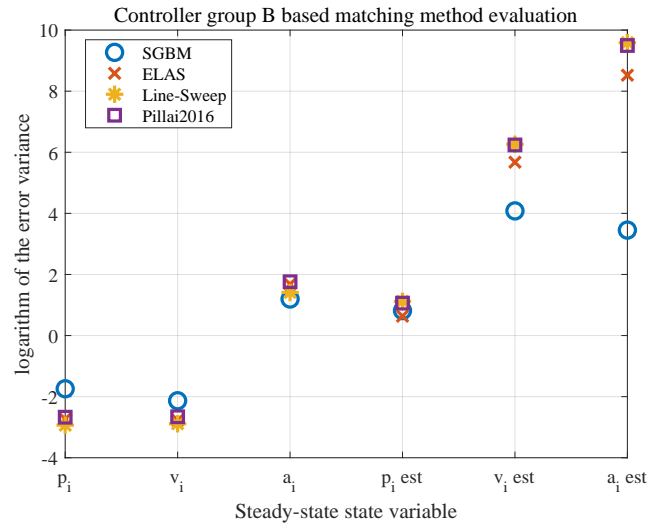


Figure 8.13 Log scale steady-state variance of each state variable with controller group B.

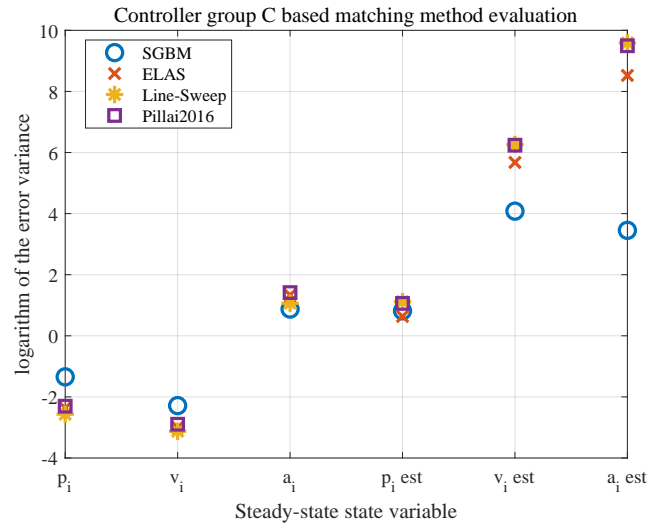


Figure 8.14 Log scale steady-state variance of each state variable with controller group C.

8.4 Conclusion

In this section, we discussed a filter design method for both stationary and adaptive observer that takes into account both the minimization of the covariance of the estimation error due to the Gaussian noise

of the sensor and the convergence speed constraint determined by the poles of the error equation.

For stationary observers, if the observer gain K that stabilizes the closed-loop characteristics is determined, the terminal value of the estimated error covariance is uniquely determined from the Lyapunov equation. The approximated optimum gain can be roughly estimated by try and error. The optimum solutions in similar situations are shown in [71] or [76], which using the property of the Lyapunov equation. This chapter focused on the matrix inequality approach so that we can take convergence conditions as simple LMI expression.

On the other hand, to minimize the estimation error covariance in an adaptive filter, a method close to the conventional Kalman Filter framework can be considered, and it is also challenging to find an exact solution. So the approximate solution can be applied in real situations.

By using these methods, it is possible to perform a design that explicitly eliminates the influence of sensor noise as much as possible while explicitly controlling the convergence speed of the observer, which is vital in high-response control.

Prospects include not only deriving a method for finding the optimal solution with low computation but also selecting from multiple sensors and expanding to multirate systems with different sensor timings [77], networked multi-sensor fusion [78] and so on.

Chapter 9

Conclusion

This paper focused on the robot automation based on the visual servo, which considers both control of robot dynamics and image sensing process to evaluate the system stability and behavior, to demonstrate integrative vision-based automation system design.

Chapter 1 shows the background in vision-based sensing and its application to an autonomous system. Recent progress in computer science and silicon semiconductor processing provides dozens of candidates of the sensor device and algorithm, so we need to choose them carefully considering its application.

In this paper, from the viewpoint of the visual servo, the concrete design guideline of the whole system, including sensing, observer, and controller, is discussed by two classifications: image-based and position-based visual servo.

The image-based visual servo described in chapter 2 to 4 can achieve precise positioning since it suppresses the uncertainty of sensing and camera modeling by incorporating the sensing part in the feedback loop. But it also has problems such as tracking performance that the orbit generation does not go as expected.

On the other hand, in the position-based method dealt with in chapter 5 to 8, there is a problem that the accuracy greatly depends on the 3D estimation, especially the distance estimation. At the same time, the control law becomes simple, since it controls in the 3D space based on the result of the 3D estimation using the image.

Chapter 2 to 4 has described displacement estimation, command value generation, and visual tracking using frequency domain information of images for motion reproduction of robot arms based on video information. In chapter 5 to 8, problems of cooperative design of controller and observer and sensor selection were solved from the viewpoint of speed-accuracy tradeoff and noise reduction for adaptive cruise control, which is a part of automatic operation based on distance estimation using a stereo camera.

In chapter 2, we proposed a 2D FFT based image displacement estimation method, which is highly accurate and false-aware, and its tuning guidelines. The performance of the proposed method was evaluated by the comparison with the conventional feature-point-based method. The proposed method is superior in estimation on vague images, constant computation time, and false-awareness.

In chapter 3, we designed an image-based visual servo system based on the features obtained from the sensing method proposed in chapter 2 and demonstrate its effectiveness in experiments. Then, the proposed coordinate transformation leads to the image jacobian to be time-invariant matrix when targeting a planar object. Therefore the feedforward control can be easily realized with proper scaling estimation, and it can greatly improve tracking performance. This visual tracking can be applied to the

Table 9.1 Relationships of each chapter.

	Image-based	Position-based	
Control	Visual tracking control with image-based feedforward. (Chapter 3)	Estimation results evaluation with adaptive cruise control law. (Chapter 7)	Noise and Convergence aware control, sensing, estimation co-design. (Chapter 8)
Sensing	Frequency domain information based image matching. (Chapter 2)	Depth estimation from height constraints. (Chapter 5)	
Estimation	Camera motion estimation from video. (Chapter 4)	Stereo sensor fusion toward wider range depth estimation. (Chapter 6)	

teaching of the operation of a robot based on the video information.

In chapter 4, we examined a method to accurately extract the camera motion from the video used in the visual tracking in Chapter 3. The proposed coordinate transformation, which is similar in chapter 3, can transform the image displacements optimization problem to simple linear least-squares. And we proposed a computationally efficient method by introducing the concept of a distance matrix. It is shown that the control law and estimator can be designed based on a suitable sensing method based on the fact that the correct weight can be obtained by the false-aware estimation proposed in chapter 2.

In chapter 5, depth estimation utilizing the height constraint of the ground vehicle and coarse/fine marker tracking that combines template tracking and color information was introduced for indoor position control of the ground vehicle. As a result, a millimeter-order-high-precision position estimation with small markers was achieved.

Chapter 6 described a distance estimation method using a stereo camera. To solve the stereo accuracy and range tradeoff, we proposed the sensor fusion technique to estimate robust and accurate depth with stereo disparity and relative scaling from monocular vision. The improvement of the responsiveness of estimation, which has been a bottleneck, is reported by proposing a switching observer using a pole arrangement instead of the commonly used extended kalman filter.

Chapter 7 discusses the evaluation of sensing and state estimator design in chapter 6 when applied to adaptive cruise control. Using actual parameters in ACC, we showed that sensor fusion based on pole placement is useful in high-gain cases where the responsiveness of distance control between vehicles is enhanced. This insists that we have to evaluate the observation not with its mean error or variance but with final control performance, including convergence speed.

In chapter 8, a theoretical calculation method using the Lyapunov equation was used to determine how sensor noise from sensors and image processing affects state estimates, state variables, and final output. This made it possible to design an estimator that reduces errors in a steady state in a control system when sensor noise and sampling are determined. In addition, by fixing the controller and the estimator, we could also select an appropriate image processing algorithm from the system information in advance.

In the image-based methods in chapters 2 to 4, it was shown that the control method is affected by determining the sensing method, and necessary estimators such as the scale amount are also affected by this. Therefore, it became clear that it is important to select the algorithm used for sensing so that it is convenient for control and estimation. The improvement of trajectory tracking performance can be expected by feedforward, which is usually difficult, with a series of visual servos based on the phase correlation proposed in this paper. Similarly, the position-based methods in chapters 5 to 8 show that the

integrated co-design of the sensing, observers, and the controller is necessary from the viewpoint of error and the convergence speed of the estimation. By applying control performance evaluation in chapter 8, we can invert the relationship between control and sensing design: that is, the desired sensing algorithms can be chosen from the control targets.

As a summary, this paper introduced an innovative framework to design control, sensing, and estimator integrally and those frameworks in this paper have relationships shown in Table. ???. Each sensing, estimation, and control method in this paper not only solves the particular problems but also applicable to other robotics application. And the methodology to choose proper combination is already explained. The robotics in the near future will require such cross-disciplinary thinking; we hope that this study will make a contribution to the development of robotics technology.

Appendix A

Homography based planer estimation

Homography matrix [10] has been used in many scenes since ancient times to estimate the state of cameras and objects. It can be used for applications such as AR (Augmented Reality) and visual servo.

Focusing on automobile applications, there are applications for estimating the ground, such as the ground and walls. The principle itself is easy to understand, but the decomposition of the Homography matrix into rotation \mathbf{R} and translation \mathbf{t} is more efficient than the principle using SVD proposed by Zhang in 1999. Theoretical advances such as [10] proposed in 2007 are progressing.

A.1 Camera projection model to homography [3]

With pinhole camera model, a 3D point at $\chi^* = (X^*, Y^*, Z^*)$ is projected on to camera image at $\mathbf{P}^* = (p_x^*, p_y^*)$ and their relationships is

$$\mathbf{P}^* = \frac{f}{Z^*} \chi^* \quad (\text{A.1})$$

, while f is focal length at pixel coordinates.

More advanced model with following camera matrix \mathbf{K} is often used for normal camera projection modeling.

$$\mathbf{K} = \begin{bmatrix} f & fs & u_0 \\ 0 & fr & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (\text{A.2})$$

In this projection model, a 3D point is projected onto the distance-normalized imaginary image plane $\mathbf{m}^* = (x^*, y^*, 1)^\top$ then transformed to actual image plane $\mathbf{p}^* = (u^*, v^*, 1)^\top$ with following set of equations.

$$\mathbf{m}^* = \frac{1}{Z^*} \chi^* \quad (\text{A.3})$$

$$\mathbf{p}^* = \mathbf{K} \mathbf{m}^* \quad (\text{A.4})$$

When we set another camera pose with the same point, $\chi = \mathbf{R}\chi^* + \mathbf{t}$, newly observed image point \mathbf{p} and the original point \mathbf{p}^* have relationship like:

$$a_g \mathbf{p} = \mathbf{G} \mathbf{p}^* \quad \left(\mathbf{G} = \begin{bmatrix} g_{11} & g_{12} & g_{13} \\ g_{21} & g_{22} & g_{23} \\ g_{31} & g_{32} & 1 \end{bmatrix} \right) \quad (\text{A.5})$$

a_g is scaling parameter and can be written with template distance Z_0 and Z ,

$$a_g = \frac{Z}{Z_0} \quad (\text{A.6})$$

We call this \mathbf{G} as homography matrix, and especially the object has a planer shape, this relationship can be applied on to whole object points.

Estimating homography often need four corresponding points and estimated from homogeneous equation. So, estimated homography $\hat{\mathbf{G}}$ has often scaling ambiguity.

To get normalized homography matrix, we need to calculate γ in $\mathbf{H} = \frac{1}{\gamma} \mathbf{K}^{-1} \hat{\mathbf{G}} \mathbf{K}$.

Normalizing parameter γ can be calculated from

$$\hat{\mathbf{H}} = \mathbf{K}^{-1} \hat{\mathbf{G}} \mathbf{K} \quad (\text{A.7})$$

$$\gamma = \text{med}(\text{svd}(\hat{\mathbf{H}})). \quad (\text{A.8})$$

and then we can disassemble \mathbf{H} to get rotation and translation with a following relationship.

$$\mathbf{H} = \mathbf{R} + \mathbf{t}\mathbf{n}^\top \quad (\text{A.9})$$

A.2 Scaling and rotation extraction

With this estimated normalized homography $G_n = \mathbf{K} \mathbf{H} \mathbf{K}^{-1}$ the scaling Z/Z_0 can be calculated from the next equation.

$$\frac{Z_0}{Z} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \mathbf{G}_n \begin{pmatrix} u^* \\ v^* \\ 1 \end{pmatrix} \quad (\text{A.10})$$

Rotation can be extracted from normalized \mathbf{H} using SVD.

Appendix B

3D Rotation and Homography Transformation

One notation of three-dimensional rotation is a notation using a 3×3 rotation matrix called Direct Cosine Matrix(DCM). There are several methods to explain 3D rotation and it depends on how the rotation axes and their order are selected. In this paper, we consider a system that rotates around the $x, y, \text{ and } z$ axes.

At this time, the angle corresponding to the rotation of each axis is expressed as $\Phi = [\phi, \theta, \psi]^T$, and each corresponds to the yaw, pitch, and roll angle. Assuming that the rotation matrix $R(\Phi)$ rotates in the order of the above roll pitch yaw, DCM can be expressed as follows:

$$R(\Phi) = \begin{bmatrix} C_\phi C_\theta & C_\phi S_\theta S_\psi - S_\phi C_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ S_\phi C_\theta & S_\phi S_\theta S_\psi + C_\phi C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi \\ -S_\theta & C_\theta S_\phi & C_\theta C_\psi \end{bmatrix} \quad (\text{B.1})$$

, while the sine and cosine of each rotation angle are expressed as S_x, C_x .

We can set some approximations for example; if small roll and no yaw, we can apply approximation $C_\phi = 1, S_\phi = 0, C_\psi = 1, S_\psi = \psi$ then, get

$$R(\Phi) = \begin{bmatrix} C_\theta & S_\theta \psi & S_\theta \\ 0 & 1 & -\psi \\ -S_\theta & 0 & C_\theta \end{bmatrix}. \quad (\text{B.2})$$

By considering situations, simpler rotation matrix can be used for estimation.

Appendix C

Relationship with H_2 norm minimization problem

The optimal gain estimation problem in 8 has a similar structure with H_2 minimization problem. This section describes an analogy with the H_2 norm minimization problem that may be helpful in minimizing the trace of the solution of the Lyapunov equation.

C.1 H_2 norm definition

The H_2 norm $\|G\|_2$ in the system of Eq. (8.2) is expressed as

$$\|G\|_2^2 = \text{trace}(D^\top D + B^\top P_0 B) \quad (\text{C.1})$$

$$\text{while, } P_0 = A^\top P_0 A + C^\top C \quad (\text{C.2})$$

or

$$\|G\|_2^2 = \text{trace}(DD^\top + CX_0C^\top) \quad (\text{C.3})$$

$$\text{while, } X_0 = AX_0A^\top + BB^\top. \quad (\text{C.4})$$

Intuitively, this is evaluating the sum of squares of the gain in the frequency domain. From Parseval's theorem, it is also the value of the root mean square of the impulse response.

C.2 LMI based H_2 optimization

Convex optimization approach such as linear matrix inequality (LMI) conditions [79] has good affinity in the sensing and controller design [80].

The condition that $\|G_d\|_2 < \gamma$ holds for the H_2 norm is equivalent to the following linear matrix inequality (LMI) condition [81].

$$A^\top P A - P + C^\top C < 0 \quad (\text{C.5})$$

$$Z - D^\top D - B^\top P B > 0 \quad (\text{C.6})$$

$$\text{trace}(Z) < \gamma^2 \quad (\text{C.7})$$

Under this LMI constraint, the lower bound of γ , which is equivalent with the H_2 norm, can be obtained by solving the SDP that solves the minimum value of γ .

This can be calculated by minimizing γ_{sq} under the following constraints in on the two variables P and Z .

$$\inf_{X, Y, \hat{A}_K, \hat{B}_K, \hat{C}_K, \hat{D}_K, Z,} \gamma_2 \text{ subject to} \quad (C.8)$$

$$\begin{bmatrix} A^\top P A - P & C^\top \\ C & -I \end{bmatrix} \prec 0 \quad (C.9)$$

$$\begin{bmatrix} Z - B^\top P B & D^\top \\ D & I \end{bmatrix} \succ 0 \quad (C.10)$$

$$\text{trace}(Z) < \gamma_{sq} \quad (C.11)$$

As a result, the estimated norm can be calculated as $\|\hat{G}_d\|_2 = \sqrt{\gamma_{sq}^*}$.

C.3 Analogy between polar constrained variance minimization estimation and H_2 minimization

The error covariance of the i_{th} state of x estimation can be written as $\text{trace}(C_j P_\infty C_j^\top)$ with standard basis vectors $C_j = (0, \dots, 1, \dots, 0) \in \mathcal{R}^{1 \times n}$ and $C_j[i] = 1$.

Therefore the estimation covariance minimization problem with stationary observer has an analogy with H_2 minimization problem. In fact, replacing some variables in Eq. (C.4); replacing A with $A - KCA$ and C with C_j , the desired optimization problem can be converted to the LMI based H_2 norm minimization problem shown in Eq. (C.4).

$$\inf_{X, W, K} \gamma_{sq} \text{ subject to} \quad (C.12)$$

$$\begin{bmatrix} A' X A'^\top - X & B' \\ B'^\top & -I \end{bmatrix} \prec 0 \quad (C.13)$$

$$\begin{bmatrix} W - C' X C'^\top & D' \\ D'^\top & I \end{bmatrix} \succ 0 \quad (C.14)$$

$$\text{trace}(W) < \gamma_{sq}, \quad (C.15)$$

with

$$A' = A - KCA, B' = K\Omega^{1/2}, C' = C_j. \quad (C.16)$$

Although the shape of this constraints such as Eq. (C.13) and Eq. (C.14) is similar to the matrix inequality, Eq. (C.13) contains two variables K and X so that it is not "linear" matrix inequality. In the H_2 norm minimization, the parameter conversion such as $Y = KX$ can be applied to linearize the equations.

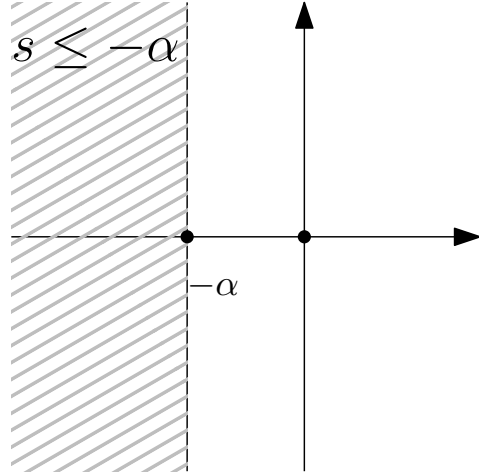
C.4 Matrix inequality based Minimum-estimation-covariance observer design under convergence speed constraints

The converted H_2 minimization problem in still do not have the convergence speed constraints.

As shown in Fig. C.1, the convergence speed condition in the continuous system that the real value of the pole is smaller than $-\alpha$ is equivalent to that the discrete poles exists in the origin-centered circles with radius $r = e^{-\alpha T_s}$ with sample time T_s .

In the LMI condition, the poles of a matrix X existing in the circle with center of c and radius r can

Continuous s plane



Discrete z plane

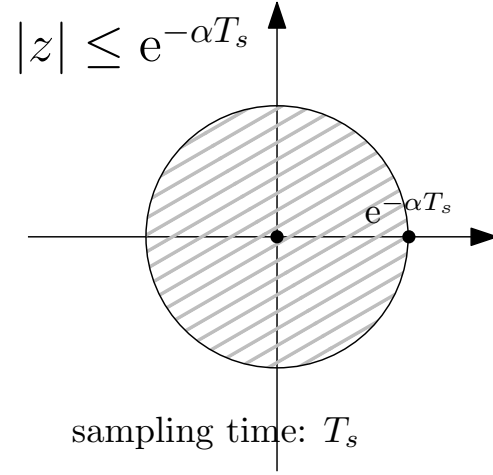


Figure C.1 LMI condition for convergence speed for continuous model and discrete model.

be written as

$$\begin{bmatrix} c(XA^\top + AX) + (r^2 - c^2)X & AX \\ XA^\top & X \end{bmatrix} \succ 0. \quad (\text{C.17})$$

So, the convergence speed constraints in LMI becomes

$$\begin{bmatrix} e^{-2\alpha T_s} X & A'X \\ XA'^\top & X \end{bmatrix} \succ 0. \quad (\text{C.18})$$

Finally, the stationary observer with the minimum estimation covariance error under convergence speed constraint can be calculated by solving the matrix inequality problem:

$$\inf_{X, W, K} \gamma_{sq} \text{ subject to} \quad (\text{C.19})$$

$$\begin{bmatrix} A'XA'^\top - X & B' \\ B'^\top & -I \end{bmatrix} \prec 0 \quad (\text{C.20})$$

$$\begin{bmatrix} W - C'XC'^\top & D' \\ D'^\top & I \end{bmatrix} \succ 0 \quad (\text{C.21})$$

$$\begin{bmatrix} e^{-2\alpha T_s} X & A'X \\ XA'^\top & X \end{bmatrix} \succ 0 \quad (\text{C.22})$$

$$\text{trace}(W) < \gamma_{sq}, \quad (\text{C.23})$$

with

$$A' = A - KCA, B' = K\Omega^{1/2}, C' = C_j. \quad (\text{C.24})$$

Since this matrix inequality satisfying Eq. (C.19) can not be easily solved, we need to try some iterative method to get approximate answer.

Appendix D

Estimation error covariance minimization via adaptive observer

D.1 Error and covariance study on discrete adaptive observer

As the same with 8, we have another approach to achieve convergence rate and optimal estimators: an adaptive observer based approach. Kalman filter, the well-known probabilistic adaptive observer, can be derived from the covariance minimization problem of its update with adaptive gain K_t at $t = k + 1$.

The updated covariance can be written as:

$$P_{k+1} = (I - K_{k+1}C)P_{k+1|k}(I - K_{k+1}C)^\top + K_{k+1}\Omega K_{k+1}^\top \quad (\text{D.1})$$

$$= K_{k+1}CP_{k+1|k}(K_{k+1}C)^\top + K_{k+1}CP_{k+1|k} + P_{k+1|k}(K_{k+1}C)^\top + P_{k+1|k} + K_{k+1}\Omega K_{k+1}^\top \quad (\text{D.2})$$

$$= K_{k+1}XK_{k+1}^\top - K_{k+1}Y - Y^\top K_{k+1}^\top + P_{k+1|k} \quad (\text{D.3})$$

$$\text{while, } X = CP_{k+1|k}C^\top + \Omega, \quad Y = CP_{k+1|k}$$

Square complete about K_{k+1} lead to

$$P_{k+1} = (K_{k+1} - Y^\top X^{-1})P_{k+1|k}(K_{k+1} - Y^\top X^{-1})^\top + P_{k+1|k} - Y^\top X^{-1}Y. \quad (\text{D.4})$$

So, when there are no constraints on the observer gain, the optimal adaptive gain to minimize P_{k+1} is

$$K_{k+1} = Y^\top X^{-1} = P_{k+1|k}C^\top (CP_{k+1|k}C^\top + \Omega)^{-1}. \quad (\text{D.5})$$

This adaptive gain K_{k+1} minimizes estimation error covariance in the step so that it is called optimal gain in the Kalman filtering for linear time invariant system.

The final value P_∞ of the covariance of the prior estimation using the stationary Kalman filter can be calculated as the solution of the following Riccati equation.

$$APA^\top - A - APC^\top (CPC^\top + R)^{-1}CPA^\top + Q = 0 \quad (\text{D.6})$$

,while R is sensor noise covariance and Q is that of the process noise.

This equation can be solved with "dare" command on matlab.

D.2 Study on adaptive gain determination with constraints on convergence

The adaptive gain calculated from Eq. (D.5) becomes less adaptable in term of the disturbance suppression.

To solve this problem we set the convergence rate constraints for the observer shown in Eq. (D.7).

$$\|e_{k+1}\| = \|(A - KCA)e_k\| \leq \alpha \|e_k\| \quad 0 < \alpha < 1 \quad (\text{D.7})$$

In Eq. (D.7), α determines the lower bound of convergence speed for this estimation.

Thus, in the adaptive filtering solution, we need to estimate the optimal observer gain which minimizing Eq. (D.4) with satisfying Eq. (D.7).

This optimal adaptive gain under convergence rate is also difficult to calculate, then we can try approximated solution such as using butterworth patterns; it is because 3-order discrete Kalman-Bucy filters have butterworth poles [82].

References

- [1] Y. Ri and H. Fujimoto, “Proposal of visual servoing using phase-only-correlation (POC),” in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, 2015, pp. 5068–5073.
- [2] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3951 LNCS, pp. 404–417, 2006.
- [3] S. Lacroix, R. Alami, T. Lemaire, G. Hattenberger, and J. Gancet, *Decision making in multi-UAVs systems: Architecture and algorithms*, 2007, vol. 37.
- [4] B. Srinivasa Reddy and B. N. Chatterji, “An FFT-based technique for translation, rotation, and scale-invariant image registration,” *IEEE Transactions on Image Processing*, vol. 5, no. 8, pp. 1266–1271, 1996.
- [5] K. Takita, T. Aoki, Y. Sasaki, T. Higuchi, and K. Kobayashi, “High-Accuracy Subpixel Image Registration Based on Phase-Only Correlation,” in *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E86-A, no. 8, 2003, pp. 1925–1934.
- [6] Manabu Omae, Takeki Ogitsu, Ryoko Fukuda, Wen-po Chiang, Cooperative Adaptive, Longitudinal Control, String Stability, “大型トラックの協調型 ACC における車間距離制御アルゴリズムの開発”, 自動車技術会春季学術講演会, vol. 44, no. 6, pp. 1509–1515, 2013.
- [7] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the KITTI vision benchmark suite,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354–3361.
- [8] S. Pillai, S. Ramalingam, and J. J. Leonard, “High-performance and tunable stereo reconstruction,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, pp. 3188–3195, 2016.
- [9] Koichi Hashimoto, “ビジュアルフィードバック制御と今後”, 日本ロボット学会誌, vol. 27, no. 4, pp. 400–404, 2009.
- [10] S. Benhimane, E. Malis, P. Rives, and J. R. Azinheira, “Vision-based control for car platooning using homography decomposition,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2005, no. April, pp. 2161–2166, 2005.
- [11] J. H. Kim and C. H. Menq, “Visual servo control achieving nanometer resolution in X-Y-Z,” *IEEE Transactions on Robotics*, vol. 25, no. 1, pp. 109–116, 2009. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs/_all.jsp?arnumber=4757222
- [12] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck, “Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle,” *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 743–749, 2009.
- [13] F. Chaumette and S. Hutchinson, “Visual servo control. I. Basic approaches,” *IEEE Robotics and*

- Automation Magazine*, vol. 13, no. 4, pp. 82–90, 2006.
- [14] G. Chesi, K. Hashimoto, D. Prattichizzo, and A. Vicino, “Keeping features in the field of view in eye-in-hand visual servoing: A switching approach,” *IEEE Transactions on Robotics*, vol. 20, no. 5, pp. 908–913, 2004.
 - [15] Koichi Hashimoto, “ロックオントラッキング顕微鏡”, 日本ロボット学会誌, vol. 32, no. 9, pp. 784–788, 2014.
 - [16] A. Gotou, H. Fujimoto, and Y. National, “Visual Servoing for Flying Object Based on Realtime Distance Identification,” *Robotics*, no. 4, pp. 0–3.
 - [17] Y. S. Hung and H. T. Ho, “A Kalman filter approach to direct depth estimation incorporating surface structure,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 6, pp. 570–575, 1999.
 - [18] R. Sabzevari and D. Scaramuzza, “Multi-body Motion Estimation from Monocular Vehicle-Mounted Cameras,” *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 638–651, 2016.
 - [19] D. Zhou, Y. Dai, and H. Li, “Ground-Plane-Based Absolute Scale Estimation for Monocular Visual Odometry,” *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–12, 2019.
 - [20] A. Saxena, S. H. Chung, and A. Y. Ng, “Learning Depth from Single Monocular Images,” *Advances in Neural Information Processing Systems*, vol. 18, pp. 1161–1168, 2006.
 - [21] S. Baker and I. Matthews, “Lucas-Kanade 20 years on: A unifying framework,” *International Journal of Computer Vision*, vol. 56, no. 3, pp. 221–255, 2004. [Online]. Available: <http://www.springerlink.com/openurl.asp?id=doi:10.1023/B:VISI.0000011205.11775.fd>
 - [22] A. Dame and E. Marchand, “Second-order optimization of mutual information for real-time image registration,” *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 4190–4203, sep 2012. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/22588592>
 - [23] D. G. Lowe, “Object recognition from local scale-invariant features,” *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2, no. [8, pp. 1150–1157, 1999. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=790410>
 - [24] C. D. Kuglin and D. C. Hines, “The phase correlation image alignment method,” pp. 163–165, 1975.
 - [25] K. Matsuo, T. Hamada, M. Miyoshi, Y. Shibata, and K. Oguri, “Accelerating Phase Correlation Functions Using GPU and FPGA,” *2009 NASA/ESA Conference on Adaptive Hardware and Systems*, pp. 433–438, 2009.
 - [26] M. a. Fischler and R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with,” *Communications of the ACM*, vol. 24, pp. 381–395, 1981.
 - [27] K. Tanaka and E. Kondo, “Incremental RANSAC for online relocation in large dynamic environments,” in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2006, 2006, pp. 68–75. [Online]. Available: <http://arxiv.org/abs/1506.07236>
 - [28] Takafumi Aoki, Information Sciences, Koichi Ito, Takuma Shibahara, Sei Nagashima, “位相限定相関に基づく高精度マシンビジョン-ピクセル分解能の壁を超える画像センシング技術を目指して-”, *IEICE Fundamentals Review*, vol. 1, no. 1, pp. 30–40, 2007.
 - [29] G. Tzimiropoulos, V. Argyriou, S. Zafeiriou, and T. Stathaki, “Robust FFT-based scale-invariant image registration with image gradients,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 10, pp. 1899–1906, 2010.
 - [30] H. Foroosh, J. B. Zerubia, and M. Berthod, “Extension of phase correlation to subpixel registration,”

- IEEE Transactions on Image Processing*, vol. 11, no. 3, pp. 188–199, 2002.
- [31] A. Alba, J. F. Viguera-Gomez, E. R. Arce-Santana, and R. M. Aguilar-Ponce, “Phase correlation with sub-pixel accuracy: A comparative study in 1D and 2D,” *Computer Vision and Image Understanding*, vol. 137, pp. 76–87, 2015. [Online]. Available: <http://dx.doi.org/10.1016/j.cviu.2015.03.011>
 - [32] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, “ORB: An efficient alternative to SIFT or SURF,” *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2564–2571, 2011.
 - [33] Y. Ri, “Fundamental Study of Vision Based Planar Object Depth and Velocity Estimation via EKF Fusion of Stereo Disparity and Monocular Scaling,” in *the 4th IEEJ international workshop on Sensing, Actuation, Motion Control, and Optimization (SAMCON2018)*, vol. No, no. 8, Chiba, 2018, pp. 8–13.
 - [34] L. R. G. Carrillo, G. R. Colunga, G. Sanahuja, and R. Lozano, “Quad rotorcraft switching control: An application for the task of path following,” *IEEE Transactions on Control Systems Technology*, vol. 22, no. 4, pp. 1255–1267, 2014.
 - [35] D. G. Lowe, “Distinctive image features from scale invariant keypoints,” *Int’l Journal of Computer Vision*, vol. 60, pp. 91–11 020 042, 2004. [Online]. Available: <http://portal.acm.org/citation.cfm?id=996342>
 - [36] Z. Chen and S. T. Birchfield, “Qualitative vision-based path following,” *IEEE Transactions on Robotics*, vol. 25, no. 3, pp. 749–754, 2009.
 - [37] A. Dame and E. Marchand, “A new information theoretic approach for appearance-based navigation of non-holonomic vehicle,” *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 2459–2464, 2011.
 - [38] E. Malis, F. Chaumette, and S. Boudet, “2-1/2-D visual servoing,” pp. 238–250, 1999.
 - [39] H. Fujimoto, “Visual Servoing of 6 DOF Manipulator by Multirate Control with Depth Identification,” in *Proceedings of the IEEE Conference on Decision and Control*, vol. 5, no. December, 2003, pp. 5408–5413.
 - [40] Y. Ri and H. Fujimoto, “Image Based Visual Servo Application on Video Tracking with Monocular Camera Based on Phase Correlation Method,” in *The 3rd IEEJ international workshop on Sensing, Actuation, Motion Control, and Optimization*, 2017.
 - [41] S. Bu, Y. Zhao, G. Wan, K. Li, Z. Liu, and J. Han, “Map2DFusion : Real-time Incremental Aerial Images Mosaic Based on Monocular SLAM,” pp. 4564–4571, 2016.
 - [42] B. Triggs, P. F. McLauchlan, R. I. Hartley, and A. W. Fitzgibbon, “Bundle Adjustment — A Modern Synthesis,” *Vision Algorithms: Theory and Practice*, vol. 1883, pp. 298–372, 2000. [Online]. Available: http://dx.doi.org/10.1007/3-540-44480-7_{-}21{}}5Cnhttp://link.springer.com/10.1007/3-540-44480-7_{-}21
 - [43] H. Park, P. H. Bland, A. O. Hero, and C. R. Meyer, “Least biased target selection in probabilistic atlas construction.” *Medical image computing and computer-assisted intervention : MICCAI ... International Conference on Medical Image Computing and Computer-Assisted Intervention*, vol. 8, no. Pt 2, pp. 419–26, 2005. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/16685987>
 - [44] N. Isozaki, D. Chugo, S. Yokota, and K. Takase, “Camera-based AGV navigation system for indoor environment with occlusion condition,” *2011 IEEE International Conference on Mechatronics and*

- Automation, ICMA 2011*, pp. 778–783, 2011.
- [45] C. C. Lin and M. S. Wang, “A vision based top-view transformation model for a vehicle parking assistant,” *Sensors*, vol. 12, no. 4, pp. 4431–4446, 2012.
 - [46] J. Lee, C. H. Hyun, and M. Park, “A vision-based automated guided vehicle system with marker recognition for indoor use,” *Sensors (Switzerland)*, vol. 13, no. 8, pp. 10 052–10 073, 2013.
 - [47] G. Yu, Y. Hu, and J. Dai, “TopoTag: A Robust and Scalable Topological Fiducial Marker System,” no. 2, 2019. [Online]. Available: <http://arxiv.org/abs/1908.01450>
 - [48] T. Ye, S. Arai, and K. Hashimoto, “An RC helicopter autonomous control system with single web camera,” *2012 Proceedings of SICE Annual Conference (SICE)*, pp. 2127 – 2132, 2012.
 - [49] J. Qian and J. Su, “Online estimation of image Jacobian Matrix by Kalman-Bucy filter for uncalibrated stereo vision feedback,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 1, no. May, pp. 562–567, 2002.
 - [50] W. Luo, A. G. Schwing, and R. Urtasun, “Efficient deep learning for stereo matching,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, pp. 5695–5703, 2016. [Online]. Available: <http://ieeexplore.ieee.org/document/7780983/>
 - [51] D. Eigen, C. Puhrsch, and R. Fergus, “Depth Map Prediction from a Single Image using a Multi-Scale Deep Network,” in *Neural Information Processing Systems Conference*, 2014, pp. 1–9. [Online]. Available: <http://arxiv.org/abs/1406.2283>
 - [52] J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” in *CVPR 2017*, 2017. [Online]. Available: <http://www.worldscientific.com/doi/abs/10.1142/9789812771728{-}0012>
 - [53] K. Okada, “ROS(Robot Operating System),” *Journal of the Robotics Society of Japan*, vol. 30, no. 9, pp. 830–835, 2012. [Online]. Available: <http://ci.nii.ac.jp/naid/10031129847/>
 - [54] G. Nützi, S. Weiss, D. Scaramuzza, and R. Siegwart, “Fusion of IMU and vision for absolute scale estimation in monocular SLAM,” *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 61, no. 1-4, pp. 287–299, 2011.
 - [55] R. B. Howard and A. I. R. F. I. O. F. T. W.-P. A. F. B. O. H. S. O. F. ENGINEERING., *Confidence Interval Estimation for Output of Discrete-Event Simulations Using the Kalman Filter*. Defense Technical Information Center, 1992. [Online]. Available: <https://books.google.co.jp/books?id=-cASDAEACAAJ>
 - [56] J. Ploeg, B. T. Scheepers, E. Van Nunen, N. Van De Wouw, and H. Nijmeijer, “Design and experimental evaluation of cooperative adaptive cruise control,” *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 260–265, 2011.
 - [57] 大前 学, “ACC と CACC のアルゴリズム”, *電気学会誌*, vol. 135, no. 7, pp. 433–436, 2015.
 - [58] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, “Speed/accuracy trade-offs for modern convolutional object detectors,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017.
 - [59] X. Y. Lu and J. K. Hedrick, “Practical string stability for longitudinal control of automated vehicles,” *Vehicle System Dynamics*, vol. 41, no. SUPPL., pp. 577–586, 2004.
 - [60] S. A. Mitchell, “Ground Vehicle Platooning Control and Sensing in an Adversarial Environment by,” *Utah State University Docter thesis*, 2016.
 - [61] S. Feng, Y. Zhang, S. E. Li, Z. Cao, H. X. Liu, and L. Li, “String stability for vehicular platoon

- control: Definitions and analysis methods,” *Annual Reviews in Control*, vol. 47, no. March, pp. 81–97, 2019.
- [62] J. Ploeg, N. Van De Wouw, and H. Nijmeijer, “Fault tolerance of cooperative vehicle platoons subject to communication delay,” *IFAC-PapersOnLine*, vol. 28, no. 12, pp. 352–357, 2015.
- [63] A. Salvi, “Cooperative Control for Vehicle Platooning: a Complex Network approach,” 2014. [Online]. Available: <http://www.fedoa.unina.it/9681/>
- [64] U. Hofmann, A. Rieder, and E. D. Dickmanns, “Radar and vision data fusion for hybrid adaptive cruise control on highways,” *Machine Vision and Applications*, vol. 14, no. 1, pp. 42–49, 2003.
- [65] S. Joshi and S. Boyd, “Sensor selection via convex optimization,” *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 451–462, 2009.
- [66] V. Tzoumas, L. Carlone, G. J. Pappas, and A. Jadbabaie, “LQG Control and Sensing Co-design,” no. February, 2018. [Online]. Available: <http://arxiv.org/abs/1802.08376>
- [67] 小野 貴彦, “観測ノイズを陽に考慮した外乱オブザーバの最適設計”, in 第 51 回自動制御連合講演会, 2008, pp. 108–109.
- [68] Atsushi FUJIMORI, Kenichi OSHIMA, “周波数依存重みを用いた LQG 制御系の設計”, 計測自動制御学会, vol. 29, no. 5, 1993.
- [69] 奥野 秀樹, 寺西 信, 渡加瀬, “観測ノイズを伴う系に対する線形関数オブザーバの一設計法”, 電気学会論文誌 C 部門, vol. 119, no. 11, pp. 1427–1432, 1987.
- [70] R. van der Merwe, E. A. Wan, and S. J. Julier, “Sigma-Point Kalman Filters for Nonlinear Estimation and Sensor-Fusion,” *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pp. 1–30, 2004. [Online]. Available: <http://arc.aiaa.org/doi/abs/10.2514/6.2004-5120>
- [71] Nakic, “Minimization of the trace of the solution of Lyapunov equation connected with damped vibrational systems,” *Mathematical Communications*, vol. 18, pp. 219–229, 2013.
- [72] M. Kytö, M. Nuutinen, and P. Oittinen, “Method for measuring stereo camera depth accuracy based on stereoscopic vision,” *Three-Dimensional Imaging, Interaction, and Measurement*, vol. 7864, no. May 2014, p. 78640I, 2011.
- [73] H. Hirschmüller, “Accurate and efficient stereo processing by semi-global matching and mutual information,” in *Proceedings - 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005*, 2005.
- [74] A. Geiger, J. Ziegler, and C. Stiller, “StereoScan: Dense 3d reconstruction in real-time,” in *IEEE Intelligent Vehicles Symposium, Proceedings*, 2011.
- [75] S. Ramalingam, M. Antunes, D. Snow, G. H. Lee, and S. Pillai, “Line-sweep: Cross-ratio for wide-baseline matching and 3D reconstruction,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.
- [76] N. Truhar and K. Veselić, “Bounds on the trace of a solution to the Lyapunov equation with a general stable matrix,” *Systems and Control Letters*, vol. 56, no. 7-8, pp. 493–503, 2007.
- [77] V. Gupta, T. H. Chung, B. Hassibi, and R. M. Murray, “On a stochastic sensor selection algorithm with applications in sensor scheduling and sensor coverage,” *Automatica*, vol. 42, no. 2, pp. 251–260, 2006.
- [78] Y. Liang, T. Chen, and Q. Pan, “Multi-rate stochastic H_∞ filtering for networked multi-sensor fusion,” *Automatica*, vol. 46, no. 2, pp. 437–444, 2010. [Online]. Available: <http://dx.doi.org/10.1016/j.automatica.2009.11.019>

- [79] P. M. Pardalos, *Convex optimization theory*, 2010, vol. 25, no. 3. [Online]. Available: https://web.stanford.edu/~boyd/cvxbook/bv_{_}cvxbook.pdf
- [80] L. El Ghaoui, “State-feedback control of systems with multiplicative noise via linear matrix inequalities,” *Systems and Control Letters*, vol. 24, no. 3, pp. 223–228, 1995.
- [81] L. Zuo and S. A. Nayfeh, “Structured H2 Optimization of Vehicle Suspensions Based on Multi-Wheel Models,” *Vehicle System Dynamics*, vol. 40, no. 5, pp. 351–371, 2003.
- [82] G. Y. Oak, “Connection between sampled-data digital filter and Kalman filter,” Ph.D. dissertation, 1972.

List of Publications

Journals

1. 李 堯希, 藤本 博志, “FFT に基づくロバストな画像のレジストレーション手法の設計” , 電気学会論文誌 D, Vol. 139, No. 1, pp.22-29, 2019
2. Eric Fujiwara, Yoshi Ri, Yu Tzu Wu, Hiroshi Fujimoto, and Carlos Kenichi Suzuki, “Evaluation of image matching techniques for optical fiber specklegram sensor analysis” Applied Optics Vol. 57, Issue 33, pp. 9845-9854 (2018)

Journals(Unpublished)

1. 李 堯希, 藤本 博志, “位相限定相関法を用いたビジュアルサーボとフィードフォワードによる軌道追従制御” , 電気学会論文誌 D, (Reviewed → Re-Submitted)
2. Yoshi Ri, Hiroshi Fujimoto, “Monocular Vision based Robust and Accurate Indoor Positioning System for Ground Vehicle with Geometrical Constraint”, RSJ Advanced Robotics, (Submitted)
3. Yoshi Ri, Hiroshi Fujimoto, “A Study on Sensor Fusion Design Analysis for Stereo Vision-based Relative Position Control”, The 6th IEEJ international workshop on Sensing, Actuation, Motion Control, and Optimization (SAMCON2020) 特集号 (To be Submitted)
4. 李 堯希, 藤本 博志, “観測ノイズを考慮した線形固定ゲインオブザーバに基づくセンサ選択手法の提案：画像ベースの車間追従制御における画像処理アルゴリズム選択問題への適用”, 計測自動制御学会 (To be Submitted)

Peer-reviewed international conference papers

1. Yoshi Ri, Hiroshi Fujimoto, “Proposal of Visual Servoing using Phase-Only-Correlation (POC) ” , The 41th annual conference of the IEEE Industrial Electronics Society (IECON2015) , pp. 5068 - 5073, Yokohama, Japan, Nov. 2015
2. Yoshi Ri, Hiroshi Fujimoto, “Image Based Visual Servo Application on Video Tracking with Monocular Camera Based on Phase Correlation Method ” , The 3rd IEEJ international workshop on Sensing, Actuation, Motion Control, and Optimization (SAMCON2017), Nagaoka, Japan, March. 2017
3. Yoshi Ri, Hiroshi Fujimoto, “Drift-free Motion Estimation from Video Images using Phase Correlation and Linear Optimization ” , IEEE the 15th International Workshop on Advanced Motion Control (AMC2018), Tokyo, Japan, March. 2018

4. Yoshi Ri, Hiroshi Fujimoto, "Fundamental Study of Vision Based Planar Object Depth and Velocity Estimation via EKF Fusion of Stereo Disparity and Monocular Scaling ", The 4th IEEJ international workshop on Sensing, Actuation, Motion Control, and Optimization (SAMCON2018), Senju, Japan, March. 2018

Domestic conference

1. 李堯希, 藤本博志, "テンプレートに対する画像の平行移動・回転・拡大縮小変位を特徴量に用いる位相限定相関法に基づくビジュアルサーボの提案", 第 33 回 日本ロボット学会 学術講演会, 1D1-04, pp.N/A, 東京 2015
2. 李堯希, 藤本博志, "画像の平行移動・回転・拡大縮小量の特徴量に用いるビジュアルサーボとその検出手法に関する基礎検討", 平成 27 年メカトロニクス制御研究会/精密サーボシステムと制御技術, MEC-15-017, pp.23-28, 東京 2015
3. 李堯希, 藤本 博志, "単眼カメラのビデオ動画に基づく動作再現を目指したビジュアルサーボの応用に関する研究", 第 34 回ロボット学会学術講演会, 1V2-08, pp.N/A, 山形 2016
4. 李堯希, 藤本博志 "ステレオ視差と単眼スケール量に基づく運動推定のための切り替え型オブザーバの提案", 平成 30 年メカトロニクス制御研究会/精密サーボシステムと制御技術, MEC-18-14, pp.29-34, 東京 2018
5. 李堯希, 川島明彦, 稲垣 伸吉, 鈴木達也, 清水修, 藤本博志 "GPS 走行データに基づいた市街地での走行中ワイヤレス給電の効果に関する一考察", 電子情報通信学会無線電力伝送研究会, pp.101-106, 京都 2018.
6. 李堯希, 清水修, 藤本博志 "柏地域での車両走行データに基づく市街地における EV の走行中給電システムの電力需要量推定", システム・情報部門学術講演会, N/A, pp.N/A, 千葉 2019

Awards

1. 平成 28 年電気学会産業応用部門大会優秀論文発表 (部門表彰)