TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Steuerungs- und Regelungstechnik

Aspects of High-Speed Visual Servo Control

Haiyan Wu

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Univ.-Prof. Dr.-Ing. K. Diepold
Prüfer der Dissertation:	
	1. TUM Junior Fellow. Dr.-Ing. K. Kühnlenz
	2. Univ.-Prof. Dr. sc. S. Chakraborty

Die Dissertation wurde am 23.04.2012 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 18.07.2012 angenommen.

Preface

This dissertation has emerged from four years of work at the Institute of Automatic Control Engineering (LSR), Technische Universität München where I stayed from 2007 till 2011.

First of all, I would like to express my gratitude to my supervisor Kolja Kühnlenz for his creative suggestions which enrich the content of my thesis, for his insightful advice which enlightens my way of thinking and for his patient guidance which encourages me to explore unknowns. I sincerely thank Prof. Dr.-Ing./Univ. Tokio Martin Buss, who gave me the opportunity to conduct the research in a very international and open working environment at one of the undoubtedly best established robotics institutes.

Special and great thanks to Prof. Dr.-Ing. Sandra Hirche for her challenging and fruitful discussions, inspiring guidance and consistent support. I would also like to thank Prof. Alexander Borst from the Max-Plank-Institute of Neurobiology in Martinsried for his guidance in neurobiology and valuable discussions. I am very thankful to all my colleagues and students for their extraordinary assistance and efforts to this thesis: Lei Lou, Chih-Chung Chen, Tian-Guang Zhang, Ke Zou and Jiayun Feng. I also want to thank Bernhard Weber for his support in translating the abstract to German. Heartfelt thanks also to all my office colleagues and friends from LSR who root for me during the time, especially for TingTing Xu, Hao Ding, Zheng Wang, Ulrich Unterhinninghofen, Andrea Bauer, Roderick de Nijs, Mohammad Abu-Alqumsan, Shushu Ma, Linlin Zhao, Hailing Li, and Wei Li. I treasure the time we spend together both in the office and in TU-Mensa. I appreciate enormously technical support from my colleagues Georg Bätz, Jens Hölldampf, Thomas Schauss, Stefan Sosnowski, Herr Jaschik, Herr Geng, Herr Gradel, Herr Kubick and Herr Stöber, as well as all the helps in administrative issues from Frau Schmid, Frau Werner, and Frau Renner.

The last and most important thanks are given to my parents and Hao. With their unconditional love and spiritual supports I could finish the PhD abroad.

Munich, April 2012 Haiyan Wu

to Hao

$\dots$

Abstract

Visual perception is one of the most important areas of vision-based robot control. The visual information is expected to be fast, accurate and reliable in providing real-time information of dynamical surroundings. Due to the limitation of computation capability, the performance of vision-based control systems, in particular of highly dynamic visual servo control systems, is often impacted by the low sampling rate of the visual feedback.

In order to overcome the low sampling rate problem, networked visual servo control, which integrates networked computational resources for parallel image processing, is considered for high-speed and high-performance vision-based control due to its merits of easy maintenance, high flexibility and robustness. In contrast to conventional visual servoing, a communication network is involved in the feedback loop for large volume image data exchange, which challenges the control design and the communication strategy. Besides, with the advances in neuroscience research biologically inspired vision system attracts increasing attention. The insect-inspired vision is computationally cheap, particularly suitable for real-time applications, and to date, however, less applied for closed loop control.

This thesis provides a comprehensive development concept for a high-speed networked visual servo control system, which merges different design issues including communication, control performance, stability, and network usage. A real-time transport protocol is developed for large volume image data transmission and thus, a cloud computing platform is established enabling high sampling rate visual feedback. A novel switching control law is proposed based on the analysis of sampled-data stochastic systems with time-varying feedback time delays. With regard to limited communication resources the cost-performance trade-off is addressed by innovative strategies of sending rate scheduling. Besides, this thesis summarizes guidelines for the design of an insect-inspired high-speed vision system for robot control. The emphasis lies on accurate motion estimation and stability of the closed loop system. A motion estimation algorithm based on a lookup table and a stable system with high feedback gains and delays are designed based on an investigation of an insect-inspired motion detector. The proposed approaches are validated by experiments conducted on seven/one degrees-of-freedom robotic manipulators. The experimental results show superior performance compared to the conventional counterpart. The work presented in this thesis contributes to the perception and control of robotics systems, and provides valuable insights for the future work in the framework of high-speed visual servo control.

Zusammenfassung

Visuelle Wahrnehmung ist eine der Hauptkomponenten für Systeme zur Roboterregelung. Von visuellen Sensoren wird eine echtzeitfähige und präzise Darstellung einer dynamischen Umgebung erwartet. Auf Grund der begrenzten Rechenleistung ist die Effizienz der bildbasierten Regelung, insbesondere der hochdynamischen Regelung, häufig durch die geringe Abtastrate des visuellen Feedbacks eingeschränkt.

Um das Problem der geringen Abtastrate zu umgehen, werden vernetzte Sichtsysteme zur Regelung verwendet, welche verteilte Rechenressourcen zur parallelen Bildverarbeitung mit ein beziehen. Dies ermöglicht eine hochfrequente und hocheffiziente bildbasierte Regelung auf Grund von einfacher Handhabung, hoher Flexibilität und Robustheit. Im Gegensatz zur konventionellen bildbasierten Regelung wird ein Kommunikationsnetzwerk in die Feedbackschleife integriert, um große visuelle Datenmengen auszutauschen. Dieser Ansatz ist eine Herausforderung sowohl für das Regelungskonzept als auch für die Kommunikationsstrategie. Außerdem gilt im Rahmen der aktuellen neurobiologischen Forschung die Aufmerksamkeit immer mehr den biologisch inspirierten Sichtsystemen. Im Bezug auf die Rechenleistung sind die Insekten-inspirierten Sichtsysteme dabei sehr effizient, was zu einer Eignung für Echtzeitanwendungen führt. Bisher werden die genannten Systeme aber noch wenig im geschlossenen Regelkreis verwendet.

Diese Dissertation schlägt ein umfassendes Entwurfskonzept für hochfrequente vernetzte bildbasierte Regelungssysteme vor. Dabei werden Aspekte wie Kommunikationsstrategie, Regelgüte und Netzwerkressourcen berücksichtigt. Ein Echtzeit-Transportprotokoll für den Austausch einer großen Menge von Bilddaten über das Kommunikationsnetz wird entwickelt. Folglich entsteht eine Cloud-Computing Plattform, die visuelles Feedback mit einer hohen Abtastrate ermöglicht. Basierend auf der Analyse abgetasteter Werte von stochastischen Systemen mit variablen Zeitverzögerungen wird ein neuartiges schaltendes Regelungskonzept vorgeschlagen. Unter Berücksichtigung der begrenzten Netzwerkressourcen wird durch die Regelung der Senderaten ein Kompromiss zwischen Kosten und Regelgüte getroffen. Darüber hinaus befasst sich diese Dissertation mit der Erforschung hochfrequenter Insekten-inspirierter Systeme zur bildbasierten Roboterregelung. Das Hauptaugenmerk liegt dabei auf der präzisen Bewegungsschätzung und der Stabilität des geschlossenen Regelkreises. Dies erfolgt auf der Grundlage einer ausführlichen Analyse der intrinsischen Charakteristika von Insekten-inspirierten Bewegungsdetektoren. Ziel dabei ist es, einen auf Look-up Tabelle basierenden Algorithmus zur Bewegungsschätzung und ein stabiles System zu entwerfen. Experimente in Form der kamerageführten Manipulatorregelung validieren die vorgeschlagenen Ansätze. Die experimentellen Ergebnisse zeigen eine signifikante Verbesserung der Performanz gegenüber klassischer Systeme. Diese Arbeit liefert grundlegende Einblicke in hochfrequente bildbasierte Regelung und bildet einen Wegweiser für die zukünftige Forschung.

Contents

Notations

Abbreviations

BMI	Bilinear Matrix Inequality
CPU	Central Processing Unit
DOF	Degree of Freedom
EMD	Elementary Motion Detector
EMDSS	Exponentially Mean-Square Stable
FPGA	Filed-Programmable Gate Array
FOV	Field of View
GPU	Graphics Processing Units
GPGPU	General Purpose GPU
IBVS	Image-based Visual Servoing
i.i.d.	independent and identical distribution
LMI	Linear Matrix Inequality
LoG	Laplacian of Gaussian
LTI	Linear Time-Invariant
LUT	Lookup Table
MJS	Markovian Jump System
MES	Mean Exponential Stability
ncRTP	Networked Control Real-time Transport Protocol
NCS	Networked Control System
NVSC	Networked Visual Servo Control
PBVS	Position-based Visual Servoing
PSD	Power Spectral Density
QoS	Quality of Service
RTP	Real-Rime Transport Protocol
SIFT	Scale Invariant Feature Transform
SURF	Speed Up Robust Features
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
ZOH	Zero Order Hold

Conventions

Scalars, Vectors, and Matrices

x	scalar/vector
X	matrix or scalar

$\dot{x}$, $\ddot{x}$	equivalent to $\frac{d}{dt}x$, $\frac{d^2}{dt^2}x$
$\|\| \cdot \|\|$	Euclidean norm
$\hat{x}$	estimated values
$\bar{x}$	mean of x

Subscripts and Superscripts

$\lambda_{\max}(\cdot)$	Maximal eigenvalue
$\lambda_{\min}(\cdot)$	Minimal eigenvalue
$(\cdot)^{-1}$	Inverse
$(\cdot)^{+}$	Pseudo-inverse
$(\cdot)^{T}$	Transpose

Networked High-Speed Vision for Visual Servoing

I	image intensity
x, y	image coordinates
$\bigtriangledown^2$	the Laplacian operator
$G(\cdot)$	Gaussian kernel
σ	standard deviation
I	number of octaves
J	number of scales
i	index of octave
j	index of scale
$r_{i,j}$	ratio between image resolution and scale parameter
R_i	image resolution of $n-$th octave
Δ	ratio difference
R	image resolution
$B(\cdot)$	Binomial distribution
$\Pr(\cdot)$	probability
l_n	number of scales in the $n-$th octave
Max_s	maximum number of scales
N_f	number of features detected in the reference image
b_n	number of features selected from $n-$th scale
d_k	sending interval
t_k	time instant
τ^{sp}	transmission delay from sensor to processing node
τ^{pc}	transmission delay from processing node to controller
τ^{ca}	transmission delay from controller to actuators
τ^{cp}	computation delay for data processing on processing node
τ^{cc}	computation delay for calculating control signals on controller
τ^{x}	image transmission delay
τ^{c}	computation time delay

$\underline{\tau}^x, \bar{\tau}^x$	lower and upper bounds of the transmission delay
$\underline{\tau}^c$, $\bar{\tau}^c$	lower and upper bounds of the computation delay
T^c	set of computation time delay
T^x	set of transmission time delay
T^{cp}	element of T^c
T^{xq}	element of T^x
D	set of sending intervals
d^m	element of D
G	maximum number of processing nodes
$\underline{d}$	lower bound of the sending interval
$\bar{d}$	upper bound of the sending interval
$\bar{\tau}^c$	the worst case processing time

Stochastic NVSC with Random Delay

t_k	time instant
I	image sequence
I_k	image captured at time instant t_k
d_k	sending interval
τ^c	image processing time delay
τ^x	data transmission time delay
τ^{c+x}	sum of processing time delay and the data transmission delay
$\boldsymbol{X}$	pose of the end-effector relative to the object
$\boldsymbol{q}$	joint displacement
$\boldsymbol{x}_0$	initial condition of the manipulator
A, B	constant matrices of appropriate dimensions
u	control signal
K	feedback gain
h	holding delay
h_k	holding interval
τ	overall feedback time delay
a	maximum number of consecutive packet dropouts
S_i	delay interval
s_i	boundary of delay intervals
$\beta_i(\cdot)$	delay-dependent indicator function
S	set consisting of T^c, T^x, and D
p_i	occurrence probability
U	number of categorized computation time delay intervals
V	number of categorized transmission time delay intervals
W	number of categorized sending intervals
s^c_u, s^x_v, s^d_w	boundary of categorized delay intervals
p^c_u, p^x_v, p^d_w	occurrence probabilities
Q, P_1	symmetric matrices
P_1, P_2	real matrices
$\mathcal{L}$	the infinitesimal generator

$V(\cdot)$	Lyapunov-Krasovskii functional candidate
e	control error
x_d	trajectory of the reference module
x_c	trajectory of the controlled module
$J(\cdot)$	cost function
C_w	communication network cost
$\bar{J}_{\text{per}}$	upper bound of performance cost
s_w	upper bound of categorized sending intervals
D_w	set consisting of s_w
r_w	boundary of tracking error interval
$\underline{Dw}$	subset of Dw
$\underline{Dw_1}, \underline{Dw_2}$	subset of $\underline{Dw}$
Pri	priority
K_p, K_d	control gain matrices

Insect-Inspired Vision for Robotics

A_1, A_2	input signals of EMD
B_1, B_2	output signals of filters
R	response of EMD
$\Delta\phi$	spatial separation between receptors
τ_L, τ'_H, τ'_L	time constants of filters
$I(\cdot)$	sinusoid signal
v	velocity
$C, A(\cdot)$	amplitude
θ_0	phase
f	spatial frequency
f_t	temporal frequency
G	transfer function
K	positive scalar
R_p	theoretical prediction of EMD response
R_a	actual EMD response
η	constant for power spectrum
T_s	spatial sampling interval
N	length of sampling points
$P(\cdot)$	power spectra
R_{all}	EMD response to original input signal
$R_{1/5}$	EMD response to input with low spatial frequency
e_R	relative response error
$\dot{\theta}_d$	desired yaw rate
$\dot{\theta}_{\text{is}}$	measured yaw rate
$\Delta\dot{\theta}$	error of angular velocity
$e_{\dot{\theta}}$	relative yaw rate error
$\dot{\theta}_{\text{emd}}$	estimated angular velocity
θ_{emd}	estimated yaw angle

u	control signal
K_p, K_d, K	control gains
M	manipulator inertia matrix
$C\dot{q}$	centripetal and Coriolis torques
g	gravitational torques
Γ	joint torques
X_c	pose of the camera in the base frame
X_o	pose of the object in the based frame
X_r	relative pose between the camera and the object
J_{img}	image Jacobian
ξ	feature points
z_c	depth
α, λ	intrinsic parameters of the camera
τ	time delay

1 Introduction

In the 1970s, visual sensor began to become a critical component of robot control systems. Driven by the applications in military, industries, and medical, such as exploration, surveillance, search and rescue, the interests in the vision-based control are growing constantly, which promotes the rapid development in both theoretical and practical aspects of the discipline. The visual information was applied in *'look-and-move'* structure from the beginning. Nowadays, it is common to implement *'visual servoing'*[1], which utilizes visual information in a closed-loop fashion for its benefits of higher accuracy and flexibility.

Visual servoing is a fusion of many different research areas, including image processing, robot modeling, control theory and real-time computing. A critical aspect of visual servoing is the perception of visual information, which is essential for maintaining the control performance. The visual feedback is expected to be fast, accurate and reliable in providing informations about the operating environment, objects, obstacles and human activities. With the advances in sensing technology and image processing theory, the performance of vision-based control systems is gradually improved. However, the performance of vision-based control systems, in particular of highly dynamic vision-based control systems, is still limited by the low sampling rate of the visual feedback caused by image processing. The problem of how to speed up visual feedback arises opens a special area in vision-based control, namely *high-speed visual servo control.* Diverse ways are available in the literature for obtaining high-speed visual feedback, e.g. through intensive parallel processing on specific hardwares such as on a light vision chip. An alternative is to utilize existing computational resources over a communication network to establish a *networked visual servo control* (NVSC) system with the introduction of networked computation into traditional vision-based control system. The resulting system exhibits several advantages. It features flexible reconfiguration capabilities, e.g. new components can easily be added. Besides, it enables simple maintenance and diagnosis with low wiring effort and digital diagnosis protocols. Standardized components and computational power can be shared among different applications, and there are no strong constraints on the components in terms of low power and low weight as typical for mobile robotic applications. It also allows for shared sensing concepts, e.g. multiple robots/applications can share sensing resources and results. In known literature some aspects of NVSC are covered, e.g. the design of data transmission, distributed computation, and networked control. However, utilizing the networked computational resources for real-time image processing which enables *high-speed networked visual servo control* is less considered.

An example of a NVSC system is shown in Fig. 1.1, in which the components, including image capturing, data processing, control algorithms and actuators, can be implemented on spatially distributed processing nodes over a communication network. The distributed sensor network provides massive sensor data, which needs to be interpreted locally and/or remotely for robot control depending on the applications. In order to perform remote data processing, the sensor data are sent to computational resources over the communi-

[1] Vision-based Control

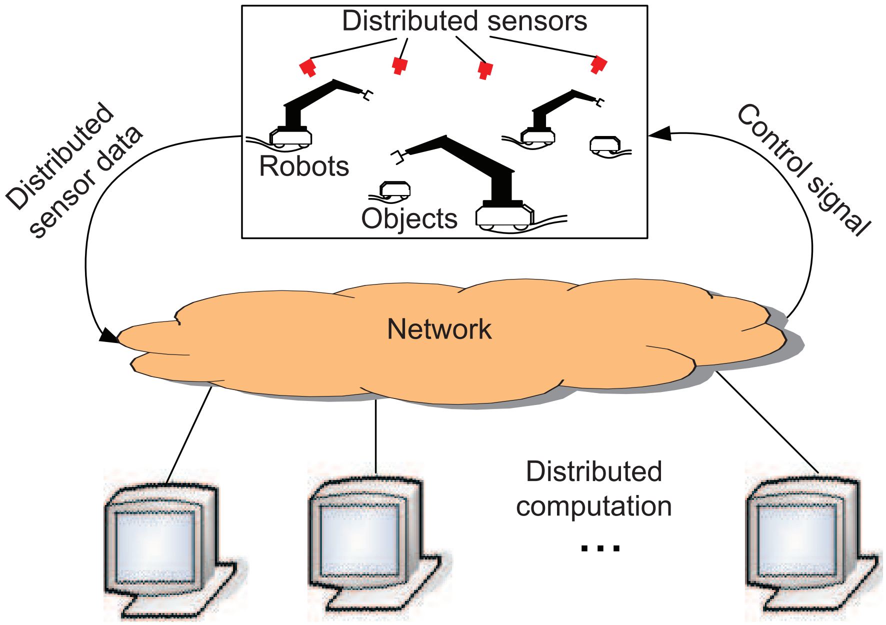


Figure 1.1: Scheme of networked visual servo control system with distributed sensors and distributed computation (cloud computing).

cation network. With two or more processing nodes available, parallel data processing can be carried out, which can speed up sensor feedback, and thus, improve the control performance.

Compared with the common visual servo control system, NVSC system has the main advantage of its improvements in processing capability, and differs in an essential way: the image processing is carried out over a communication network. This characteristic results in several fundamental issues on data transmission, controller design and cost-performance trade-off, which are less considered in known literature. There are various data transport protocols in the public domain. However, how to transmit massive image data along with ensured real-time property has not been addressed yet. Some transport protocols can only support the transmission of packages with small size, while some others may have unpredictable transmission delay that is critical to visual servo control system. In addition, the feedback delay, especially the stochastic delay due to image processing and transmission, and the time-varying image sending interval have been largely ignored in controller design in all other works. Moreover, the trade-off between the control performance and the cost has not been explicitly exploited for visual servo control systems.

Apart from the network-based approach for high-speed vision, the research in the field of neurobiology, especially the fly's visual system, inspires the buildup of an *insect-inspired high-speed vision* system for robot control. Fly possesses a tiny brain, which contains neurons one hundred thousands times fewer than that of a human brain. The fly's brain performs delicate motion feats, such as tracking and chasing the conspecifics, navigating through tunnels or flying between obstacles, and controlling their flying speed. These tasks require a computation time of only a few milliseconds, which greatly preponderate over the state-of-the-art robotic systems. Therefore, the inspiring manifestations of the fly's visual system make it one of the most interesting research domains for both neuroscientists and engineers. A novel approach comes up based on a biological model–*elementary motion* *detector (EMD)*. Motion perception based on the fly's vision system is computationally cheap and, thus, particularly suitable for real-time applications. The *Reichardt-detector*, which is well known as the model describing the process of local motion detection in the fly at algorithmic level, is implemented for different purposes in the literature. However, the utilization of EMD for motion estimation in particular for accurate angular velocity estimation is still in controversy due to its high sensitivity to the features of the input signals, e.g. the contrast and the spatial frequency of the input images captured by the camera.

The main objectives of the work presented in this thesis are the design of real-time transport protocol for massive image data transmission, the investigation of controller design, stability analysis and sending rate scheduling for cost-effective NVSC systems, and the design of motion estimation algorithm based on EMD for high-speed insect-inspired visual servoing system as well as the stability analysis.

The main challenges to the development and the design of NVSC systems are fast and reliable visual perception, control law stabilizing the closed loop system, and cost-performance trade-off. The challenges to the design of insect-inspired visual servo control are reliable motion detection with elaborated EMD and accurate motion estimation. These challenges are summarized in the following.

1.1 Challenges

The challenges of designing networked-based and insect-inspired high-speed servo control systems rise from several aspects related with efficient data transmission, high-speed image processing, sensor data fusion, cost-effective controller design, and stability issues. Some of the key issues of the design and control of high-speed visual servo control systems targeted in this thesis are summarized as follows:

Vision Issues

A common vision-based control system possesses typically a vision system of 30 Hz because of the limitations of the hardware and image processing algorithms. Fortunately, on the hardware front, cameras and frame grabbers with high performance characteristics, e.g. high frame rate, large image resolution, and high bandwidth communication link, are available nowadays. It is also feasible to put sophisticated image processing on relatively low-cost chips, which are, however, fixed and have special purposes. On the algorithm front, most work in the literature concentrates more on the robustness of image processing, but less on the computation time. With increased complexity in the algorithm, the computation time is considerably increased. It is well-known that even a small time delay can result in control performance degradation, and may even lead to unstable closed loop behavior. Therefore, in order to achieve improvements in stability and control performance, it is desirable to increase the sampling rate of the visual information, which can be achieved by parallel image processing on computational resources available over the communication network. In order to further reduce the computation time on individual processing node within the networked computation utility, advanced hardware implementation and simple image processing algorithms can be applied.

In this thesis the main challenge of networked vision system is an increase of the sampling rate and the robustness of visual feedback. Another challenge is the design of insect-inspired vision for high-speed motion perception. The aspects towards real-time transport protocol for large volume image data transmission, features of the processing platform, novel velocity estimation based on EMD are addressed.

Control Issues

For common visual servo control systems, several control schemes are identified, such as position-based visual servoing, image-based visual servoing, and 2-1/2 D visual servoing. The advantages and the disadvantages of different control methods are intensively studied in the literature. However, during the design of control law, not much attention is given to the time delay in the feedback loop. Processing image data results in a long computation time delay. In contrast to conventional feedback control loops, the transmission of sensor data over a communication network within a NVSC system introduces uncertainties such as (time-varying random) time delay. The influence of the feedback delay on the performance and stability of the visual control system is omitted to a great degree in the design of the conventional visual servo control systems. There are available techniques dealing with the communication time delay in conventional networked control systems (NVCs). However, those cannot be directly applied to NVSC, as the computation time delay is different from the transmission delay, due to the data-dependent conditional branches/loops during image processing. Besides, the sensing of image data may be triggered in different ways, e.g. periodic or aperiodic sensing, which also needs to be considered in the control design.

The main challenge of control design is the investigation of a promising control approach, which guarantees the control performance and maintains the stability of the closed loop system regarding the time-varying feedback time delay.

Communication Related Issues

In contrast to most conventional NCSs, the data transmitted within a NVSC system are large volume image data. It is considered to be beneficial to transmit a whole image fame over the communication network as the most time consuming part of image processing is feature extraction from the original images. The transport protocols in the public domain have either limited package load or unbounded transmission delay. Therefore, they are not suitable for real-time image data transmission in vision-based applications. Moreover, high communication network load results in large communication time delays and high packet drop rates. A communication network beyond the channel capacity results in a complete communication blackout. The communication constrains are critical in particular in wireless communication networks. Therefore, in the presence of limited communication bandwidth, keeping the data transmission over the communication network at a minimum under the constraint of sufficient control performance becomes a challenging task.

The main challenges are the design of an appropriate real-time transport protocol for large volume image data transmission satisfying real-time characteristics, and the exploration of scheduling strategies coping with the cost-performance trade-off for NVSC systems.

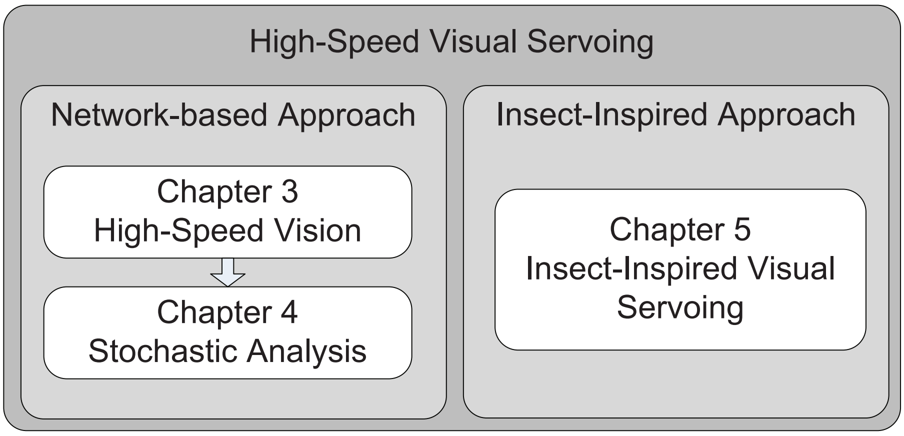


Figure 1.2: Outline of the dissertation.

1.2 Main Contribution and Outline of the Thesis

The primary goal of this thesis is to design the NVSC system, and the insect-inspired vision system, which both outperform the conventional approach in terms of high-speed visual feedback. A NVSC system utilizes the existing computational resources over a communication network for parallel image processing, which enables high sampling rate visual feedback. Instead of simple image processing, advanced image processing algorithms can be implemented to obtain more reliable and accurate perception results for high performance robot control.

In order to design a NVSC system, the fundamental issue is to realize *real-time image data transmission*. Transmission of large volume image data at high frequency in real-time is studied in this thesis. In order to exploit the benefits of NVSC, aspects towards *stability analysis* and *control design* are investigated. Moreover, in order to transfer the biological results to technical applications, the fundamental properties of the *inspect-inspired motion detector* are analyzed and utilized to estimate the motion from the image sequence at high speed. The structure of this thesis is highlighted in Figure 1.2. The main contributions of this work are presented in the following.

Cloud Image Processing for NVSC

The available techniques accelerating image retrieval and image processing on specific/advanced hardware in the literature restrict usually to their applications. The existing networked computational power has not been considered for real-time parallel image processing. Distributing the images captured by a high-speed camera to the processing nodes over the communication network, and processing the image data in a parallel manner lead to high-speed visual feedback. Different from the existing methods which mainly use massive parallel processing and rather basic image processing, the combination of high performance image processing and parallel processing mechanism over the communication network is rather new. In Chapter 3 the details about the design of the *real-time image transmission protocol* and the *cloud computing platform* are presented. Besides, a novel *robust image processing algorithm* is proposed to achieve reliable image processing results.

Stability Analysis and Control Design for NVSC

With the parallel image processing platform established in Chapter 2, it is focused in Chapter 3 on the stability and control issues of the NVSC system. The transmission of image data over the communication network results in unavoidable transmission time delay. Additionally, the computation time delay caused by image processing exists in the feedback loop. The time delay in the feedback loop degrades the control performance, and potentially even renders the system unstable. The conventional control approach which approximates the delay by its upper bound results in robust but conservative design. In Chapter 4, a *switching control* approach is proposed, which switches depending on the size of the feedback time delay. Considering the limited network bandwidth, optimization of the trade-off between the control performance and the network cost is another focus in this thesis. Novel *sending rate scheduling* strategies are designed to reduced the network load caused by image data transmission, combined with guaranteed control performance.

Insect-Inspired Motion Vision System

After solving the design problems of networked high-speed visual servoing in Chapters 3 and 4, Chapter 5 focuses on an alternative approach of high-speed vision inspired by the fly's vision system. The insect-inspired motion detector, which mimics the mechanism of motion interpretation in fly's brain, is utilized in closed loop visual servo control. In order to utilized the insect-inspired motion detector, its intrinsic characteristics are analyzed firstly. Since the raw motion detector cannot be directly used due to its low accuracy in motion estimation, upgrading the raw motion detector to a quantitative estimator becomes the first task of the system design. Different from other works using artificial stimuli as input, in this thesis it is focused on using real-time image sequence captured in an unconstructed environment. A *lookup table based approach* together with an *elaborated motion detector* are proposed to improve the performance of motion estimation. In addition, the stability of the closed-loop system is analyzed by looking into the response-velocity relationship of the motion detector.

The dissertation is concluded in Chapter 6 with a summary and a discussion about future work.

The aspects addressed in this thesis contribute to the generic design of networked visual servo control and a fundamental understanding of insect-inspired vision. The presented frameworks integrate the networked computation resources into the conventional visual servo control, and transfer neuroscience techniques to biologically inspired strategies for robot motion control. In highlighting multi-disciplinary perspectives of perception and control, this thesis makes several noteworthy contributions and serves as a based line for future studies in this area. A variety of applications and examples are presented to highlight the benefits of the integrated NVSC and the insect-inspired motion vision system.

2 Related Work

Vision-based robot control has been extensively studied in the past decades, where much attention has been focused on the development of sensing technology, image processing techniques and control scheme design [1]. Surveys of vision-based robot control can be found in [2–4], which give an overview of different visual servoing techniques with performance comparisons. Originating from the 'camera to hand' and 'camera on hand' configurations, the sensing capability is enhanced nowadays with more complicated and intelligent camera setup, e.g. networked cameras, camera arrays, and camera skins [5–7]. In order to achieve high control performance of vision-based control system, fast and robust image processing algorithms are of great importance. The development of image processing theories, e.g. applying image moments, Laplacian of Gaussian, and wavelets [8–10] for feature extraction, promotes the design of high-performance image processing algorithms. However, the long image processing delay, in particular the delay caused by running advanced image processing, is a long standing problem, and it becomes a bottleneck of developing high-performance visual servo control systems. With the recent advances in computation and communication technologies, parallel computation based on networked computational resources (*cloud computing*) has gained more and more interests for high-performance computing [11–14]. In addition, the evidence from the research in neurobiology shows that the vision system of insect possesses simple mechanism of motion detection, which motivates the development of insect-inspired high-speed vision system [15–17].

In this chapter the development of different aspects of visual servo control is reviewed. Firstly, the techniques available in the literature for accelerating visual feedback are discussed. Secondly, an overview about the data transmission for real-time vision system is given. Thirdly, different methods of system modeling, control schemes and scheduling strategies are surveyed. Finally, the state-of-the-art approaches related to the insect-inspired motion detection are discussed.

2.1 High-Speed Vision System

For high-performance vision-based robot control, it is desirable to process the image data in a fast and robust manner to provide necessary and reliable perception results for the following control step. The vision systems in literature employ different type of devices in image processing, such as microcontrollers, general processors, application specific circuit (ASIC), digital signal processors (DSPs), and reconfigurable hardware, e.g. field-programmable gate arrays (FPGAs). The image data captured by the camera can be processed either locally or remotely, depending on the computation capability of the system in various applications. According to the manner of image processing, the structure of image processing is basically divided into two types, see Fig. 2.1. The most general architecture is to connect the camera to a normal personal computer (PC), which runs image processing algorithm locally, as shown in Fig 2.1 (a). Since a normal PC usually has limited computation power, it often results in low sampling rate visual feedback. In

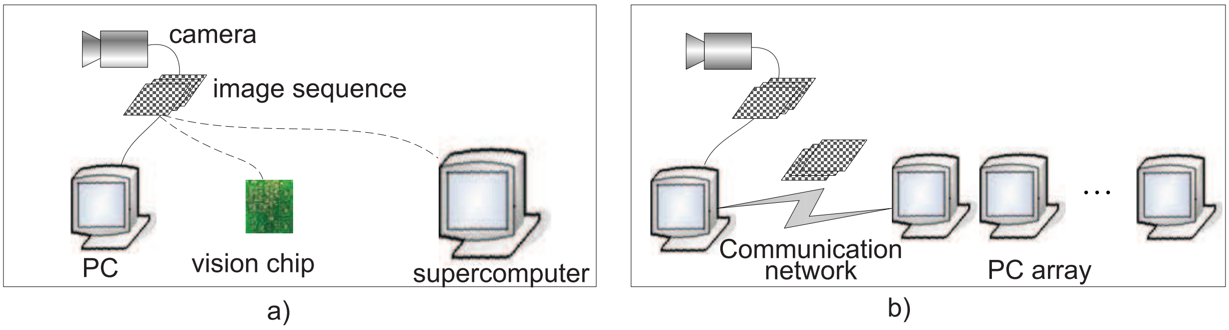


Figure 2.1: Image data processing locally or/and remotely. a): Local image processing on a normal PC / specified hardware (e.g. vision chip) / supercomputer. b): Remote distributed computation on PC array consisting two or more PCs.

order to accelerate image processing, specified hardware such as FPGA, graphics processing units (GPU), and smart vision chip, are adopted for image data processing in some applications. Another way to deal with image processing is to send the images to a supercomputer (see also Fig. 2.1 (a)), which outperforms in superior computation capability. Considering the existing computational power over the network, it becomes more and more popular to distribute images over the communication network for parallel processing. In Fig. 2.1 (b), an array of PCs over the network is utilized for cloud computing. Compared to the single-PC structure in Fig. 2.1 (a), the computation platform in Fig. 2.1 (b) has more computation power. In the following the work about high-speed vision systems is reviewed. According to the platform chosen for image processing, they are divided into two parts: i) high-speed vision on advanced hardware, and ii) high-speed vision on networked computation resources.

High-Speed Vision on Advanced Hardware

In literature many different vision systems are developed, and much research is done towards high-speed visual feedback with massive parallel processing. Vision chips capable of processing image data at 1000 fps have been developed. For these chips, the sensing units and the processing units are integrated on single chips [18–21]. For example, an artificial retina with processing facilities is built on a complementary metal-oxide-semiconductor (CMOS) circuit in [18]. In [19] a simple target tracking algorithm is implemented on a very-large-scale integration (VLSI) chip. A flexible vision chip with single-instruction multiple-data (SIMD) image processor is designed in [21] to perform more advanced processing. However, these vision chips can only run for images at low spatial resolution. Currently for computer vision the images captured by a camera typically have an image size of 640 × 480 pixels. The vision chips mentioned above cannot be applied for general-purpose image processing.

More recently, in order to accelerate visual feedback, advanced hardware such as FPGA, which is reconfigurable and flexible, is devoted to the design of vision system . Much effort is done to build vision system on FPGA for robot control [19, 22, 23]. By using FPGA, some requirements of processing image data for robot control are met: high-speed parallel processing, sufficient computational power and low cost. For example, the parallel processing capability of FPGA is utilized to build a vision system for image feature extraction in 1 ms [22]. A vision system of 955 fps is also implemented on FGPA for shape measurement in [23]. An optical-flow processing algorithm working as virtual motion sensor is implemented on FPGA [24]. Recently, an intelligent high-speed vision system is built, which is able to process a image with a size of 1024 × 1024 pixels at 1000 fps [25]. For robot control, the speed of sensory feedback plays an important role. On the other hand, the accuracy of the processing results obtained from the sensor data is also critical. However, with FPGA-based vision systems mainly rather basic/low-level image processing algorithms are implemented [26, 27]. Besides, compared to the CPU implementation with a real time operation system (RTLinux), the complexity towards software implementation on FPGA also limits its usage for general-purpose applications, as discussed in [28].

In order to reduce the computation-intensive workloads, the graphics processing units (GPUs) is another powerful computational hardware for image processing. With the development in graphics architectures, the graphics hardware gets faster, and the computational capability is improved accordingly. Therefore, it becomes a cost-effective computation platform. Highly parallel computation and increased memory bandwidth make GPU a strong candidate for high-level image processing. Accelerating a wide range of applications with GPU leads to work in general purpose GPU (GPGPU) processing. A detailed overview about GPGPU implementations is given in [29]. It is focused here on the work done in GPU vision and imaging. For example, early work in [30] mapped wavelet decomposition and reconstruction onto GPU, which reduces the lag in visualization cycle. A real-time system for 3D depth estimation running on a NVIDIA GeForce 4 graphic card creates depths maps quickly [31]. In [32], it achieves a 2.8 speedup factor against the P4 system by mapping both the motion estimation and visualization algorithm onto graphic cards. In recent years, the compute unified device architecture (CUDA) [33] is developed for allowing programming on GPU hardware with no knowledge requirement on the graphic pipeline, and thus, increases the flexibility in mapping operation onto GPU hardware. For instance, the optical flow computation proposed by Horn and Schunk [34] is implemented on CUDA [35]. A multi-GPU implementation of saliency map approaching a frame rate of 313 fps for images of size 640 × 480 pixels, is realized on four NVIDIA GeForce 8800 graphic cards [36].

Nowadays, benefiting from the advances of the multi-core computing technology, e.g. the Intel Cell [37], Niagara [38], and Larrabee [39] for highly parallel algorithms, the real-time performance of the image processing algorithms is greatly improved [40]. In [41] it is shown that with the optimized workload on a 32-core system the body tracking algorithm achieves a speed-up factor of 26. An efficient hand gesture tracking system is realized on a processor with eight cores, whose operating speed is almost 13 times faster than the single-core processor approach [42]. Because of the introduction of multi-core processors, the performance of the image processing algorithm is significantly improved, e.g. the implementation of Hough Transform on a 8-core machine achieves 25% better performance compared to that on single-core processor [43].

The performance of the implementations on FPGA, the general-purpose CPU, multi-core CPU and GPU are compared and evaluated in [44–46]. In [44], 3D tomography is computed with FPGA virtex 4, or a general purpose processor including a Xeon dual core and a Pentium 4, or with graphic cards NVIDIA 8800 GTS. The computation time is largely reduced benefiting from the parallel computation capability of these processing units. The GPU outperforms the other platforms, e.g. it is 50 times faster than a Pentium 4. In [46] a set of throughout computing kernels, e.g. convolution and FFT (fast Fourier transform), is implemented on Intel Core i7-960 and NVIDIA GTX280. The CPUs and GPUs show closer performance than that reported in other studies. It is due to several factors contributed to the performance such as the type of CPU/GPU and the optimization algorithm for the code.

The two major computation platforms CPU and GPU are suitable for different applications. With the emergence of fusing the functionality of the CPU and GPU, hybrid CPU-GPU platforms are used to accelerate the computation, e.g. the CPU-GPU platform for the computation of the matrix inverse in [47], and the fused CPU+GPU processor for reduction/FFT calculation in [48]. In addition, powerful computer system (supercomputers), e.g. K computer in Kobe, Japan and Tianhe-1A in Tianjin, China [49], are installed for high-performance computation. However, the cost and the maintenance difficulty associated with these systems limit their wide-spread adoption.

High-Speed Vision on Networked Computational Resources

With the development of computation and communication technologies, parallel computation based on networked computation resources (cloud computing) has gained more and more interests for high-performance computation [12–14, 50, 51]. Different from concentrating on developing either advanced hardware (e.g. FPGA, GPU and supercomputer), or parallel version of algorithm, the utilization of the computing resources interconnected by one or more communication network for parallel and distributed processing becomes a promising solution to the increased computing requirements in future. For the typical computing environments, data processing is carried out locally, while with the networked computing resources it is conducted remotely. A significant advantage of the network computing system is that it utilizes existing general purpose processing nodes, and thereby it requires little or no additional investment [52]. The development of data communication and network technology, e.g. communication protocols and communication media, enables data processing over the communication network in a straightforward and efficient manner.

The data processing of robot control can be represented by different levels such as such as multi-robot level, robot level, functions level and so on, which can be implemented in parallel computation manner [12]. In this thesis, it is focused on algorithm level, in particular the image processing algorithm. An example of utilizing a network of workstations for parallel image computation is given in [52]. The computation environment is evaluated with some low-level image processing algorithms such as edge detection and template matching [53], and it presents efficient speedup in computation time. Another example of image processing on a network of workstations is presented in [14]. A testbed consisting of high-speed workstations is designed for image processing (edge detection with Sobel operator and the time performance is studied [54]). It is concluded that the divisible load theory outperforms the equal-partitioning strategy in tasks scheduling among available processing nodes.

In [55] distributed computing approach is proposed for geographic systems requiring high-performance computing. Cooperative use of the networked data storage and data processing capabilities provides an economic and efficient computing platform for applications with large-scale source data.

By exploiting the computational power of individual workstation massive parallel and distributed computing over the communication network becomes a primary choice for solving the computation-intensive problem in future. However, coming to the practical implementation it confronts challenges of meeting the criteria of connectability (e.g. ensuring network security, efficient data transmission), performance (e.g. balancing the load among workstations, control design considering communication time delay), and cost (reducing communication cost regarding limited communication bandwidth) [56]. As pointed out in [57] the central design problems of distributed computation systems are *'what to compute'* and *'what to communicate'*. To answer these two questions, different computational power of each processing node and the communication bandwidth should be considered. In order to balance the load of processing nodes and to utilize the networked computing resources efficiently, suitable balancing strategies which optimize the utility of the computing capability of each node and minimize the inter-processor communication are desired. The old-fashioned strategy of distributing loads evenly results in poor performance for heterogeneous networked systems. Therefore, fast, robust, and efficient scheduling strategy considering the loads of the processor and the network is necessary. Much research effort is devoted to the design of promising scheduling mechanisms in this research domain [58–61].

2.2 Real-Time Image Data Transmission

Benefiting from distributed computation the image processing is accelerated, and thereby a better control performance is achieved. However, the communication bandwidth constraint of the networks, in particular of the wireless networks, challenges the design of the distributed computation platform. A key issue of distributed image processing is the transmission of image data in a fast, robust and efficient manner. In order to solve this problem, research effort is devoted to the development of appropriate image transmission strategies [62–67].

In visual sensor networks, e.g. for environment monitoring and target tracking, data aggregation algorithm, which fuses data from multiple sensors, is employed to reduce the size of transmitted image data [62, 68, 69]. Aggregating image data requires the design of a proper mechanism for aggregation. For example, in [63, 64] image data compression and transmitting in energy constrained wireless network is studied. The total energy consumption is concerned with the required image quality during the image compression process. In order to save the communication bandwidth, the image data from different cameras in a camera sensor system is compressed (distributed compression) in [70]. The geometrical information about the locations of the camera and the targets of interests is applied to correlate the image information captured from different view angles. Moreover, the useful temporal correlation among the images is applied in data transmission in [71]. It runs image processing locally before sending it to a monitoring center. Only the regions containing targets of interests are transmitted, and thus, the overall energy consumption is reduced.

Along with the consideration of energy consumption, for applications of real-time vision-based control a key factor during the design of image transmission is the transmission delay. The time delay in the feedback loop degrades the overall control performance, and even renders the system unstable. Therefore, the design of a real-time image transmission protocol with low latency is an important yet a challenging task. For Internet-based data transmission images are usually transmitted by lossless protocols such as TCP/IP, which triggers retransmission of data caused by packet loss. In order to overcome the lengthy image transmission delay due to retransmission mechanism, a fast lossy Internet image transmission scheme is proposed in [72]. It concentrates on image bits devoted to error correction during image compression, and the retransmission delays are thus eliminated. With the designed scheme, the image transmission is five times faster than that with TCP/IP. In [73] a low-delay wireless network is built for video transmission. A video proxy server is added on the transmission path to manage the real-time delay constraint. In order to transmit video over the Internet a TCP-friendly transport protocol is adopted in [74]. The transmission latency in TCP is reduced by filtering a scalable video bit-stream.

The other technical issues about transmitting image over network such as congestion control and error control from the compression perspective are not the focus of this dissertation. Readers interested in these issues are referred to [75] for an overview.

For high-speed transmission of image data from sensors to the processing nodes, a proper physical layer has to be built. Nowadays, the wired Internet (1000BASE-T Ethernet) is widely used in various applications, as in the examples introduced above. Other data link/physical layer devices such as EtherCAT (Ethernet for control automation technology) [76], CAN (controller area network) [77], SERCOS [78] or WLAN (wireless local area network) [79], either need dedicated hardware or have limited transmission rates, which thus limits the extension for its wide-spread use in robot control. For example, EtherCAT and RTNET based on Ethernet have deterministic delays. However, the overhead for synchronization (EtherCAT: token ring, RTNET: TDMA) introduces high bandwidth penalty, meaning that the average transmission delay is very significant. By WLAN, the theoretical speed of 802.11n is 300Mbps. However, the maximum real-world speed is only about 100Mbps, which limits the transmission of images (640×480 pixels) with rate higher than 60 Hz.

2.3 Vision-based Control

Before introducing the state-of-art of NVSC, vision-based control is firstly briefly surveyed in this section. At the very beginning vision-based control performs in a 'look-and-move' manner which indicates a open loop control fashion. It is nowadays more common to implement closed loop vision-based control with the benefits of improved control performance compared to the open loop approach [1]. In the following sections selected basic concepts of visual servoing are briefly introduced. A survey of the state-of-the-art approaches on several important aspects relevant to this thesis is presented.

Image-, Position-based, and Advanced Approaches

There are typically two visual servoing schemes classified according to the task space, the *image-based visual servoing* (IBVS) in image space and the *position-based visual servoing* (PBVS) in Cartesian space [3, 80, 81].

By the image-based approach, the camera is usually mounted on the end-effector of a manipulator, commonly referred to as *eye-in-hand* configuration. The control objective is to drive the manipulator to a desired position and rotation angle such that the features in the image coincide with that in the image captured a priory. Since image features are described in image space, and the manipulator control is performed in joint space, the image Jacobian [82] is thus introduced to map the relationship between the features motion in image space and the camera motion in Cartesian space. The major advantage of image-based approach is that it is robust to calibration error. However, it also has several drawbacks in practical implementation. For example, the depth information, which is difficult to estimate from the monocular camera system, is needed for calculating the Jacobian matrix. Besides, the image-based control approach could not guarantee the selected features always staying within the field view of the camera. Losing the features in the image plane may result in control failure.

By the position-based approach, camera can be either mounted on the manipulator or fixed in the workspace (*'eye-to-hand'* configuration). The objective is to drive the manipulator to a pre-defined pose with respect to the background of the workspace or to a stationary/moving target. Therefore, the control error is defined in the Cartesian space. An advantage of this approach is that it is possible to do trajectory planning in Cartesian space rather than in image space. The relative pose is recovered from the image data based on the camera calibration results. In contrast to the image-based approach this method can be susceptible to calibration errors.

In order to compensate for the potential shortcomings of these two approaches it comes to hybrid control scheme which combines the merits of the image-based approach and the position-based approach in visual servo control [83–86]. For example, Mails et al. proposed a *2-1/2-D control* law which decouples the six degrees of freedom (DOF) of the camera. With this method the 3-D model of the target is not needed, and it is more robust to camera calibration error compared to the position-based approach, with, however, the cost of being more sensitive to image noise compared to the image-based approach. With the partitioned approach the control performance is optimized by assigning distinct controllers to individual degrees of freedom. Another way to optimize the control performance is to use a switching control law which switches based on the criteria of the Lyapunov function [87]. The switching controller switches between the IBVS controller and the PBVS controller considering the thresholds of the Lyapunov functions. With appropriately selected switching thresholds, a better control performance is achieved compared to the IBVS or PBVS approach.

Perception Issues

Sensor techniques and perception capability play important roles in the design of robust visual servo control systems. Since visual sensors provide the real-time information of dynamical surroundings for system control, they are expected to be low-cost, low power consumption, fast and accurate. With the development of sensing technology there are various types of sensors available for different applications. For example, an ultra light camera 'Firefly MV' from Point Grey [88] is mounted on a quadrotor in an air-ground multi-robot system [89]. A high-speed camera MC1311 from Microtron [90] is connected to a FPGA board for optical flow detection [236]. A low-cost depth sensor Microsoft Kinect [92] is applied to establish an inexpensive robot platform for teleoperation [93]. In addition, frame grabbers with high performance characteristics and high bandwidth communication link are available nowadays which enables the transmission of high resolution images with high frame rate from sensors to the receivers for image processing. Apart from the monocular and binocular camera system, more advanced sensing platforms such as networked cameras [5, 94], camera arrays [6, 95, 96], and dense skins of camera (multi-camera eye-in-hand systems) [7] are developed to meet the requirements of different applications.

In order to obtain useful information from the captured images, e.g. the locations of the targets and the obstacles, the human activities, and the information of the operating environments, the first step is to run low-level image processing algorithms performed on pixel-level, such as feature (e.g. corners and edges) extraction from the images. With the results of low-level image processing high-level image processing is then followed to extract the information such as 3D location of the target for robot control. Model-based approach is another class of tracking objects. It uses models of the tracked objects such as CAD model or a 2D template of the objects [97–99]. The development of image processing techniques for target tracking helps to improve the accuracy of image processing results, which is critical to overall system control performance and stability. In this thesis, the feature-based approach is also adopted for target tracking.

The feature detection algorithm, namely scale invariant features transform (SIFT) [9], is known for its robustness to wide illumination variance and view angles changes. SIFT is widely used for feature extraction in the domain of robot control [100–102]. Later on, Bay et al. proposed another novel feature extraction algorithm, namely speeded up robust features (SURF) [103], which outperforms the existing methods in terms of its distinctiveness, robustness and speed. SURF-based application examples can be found in [104–106]. In order to achieve real-time performance, GPU implementation is commonly adopted to speed up the algorithms [107,108]. A comprehensive overview of local invariant feature detectors can be found in [109,110].

Stability Issues

In the literature of visual servoing the main focus of control design is the kinematics of the manipulator. The dynamic issues due to the nonlinear robot dynamics and visual sensor are essential to high-performance control. In [111] the nonlinear robot dynamics is considered in controller design, and an asymptotically stable system is yielded with the designed controller. The system is capable to cope with the uncertainty in the camera parameters such as lens distortion. The problem of uncertainties in camera and robot dynamic model parameters is also addressed for tracking of moving targets in [112]. An adaptive visual servo control approach is proposed using a Lyapunov-based design.

For dynamic performance analysis another important factor is the feedback time delay. However, less effort has been spent on the control design for visual servo system with feedback time delay. Compared to conventional motion control systems, the additional time delay caused by image processing leads to difficulty in the control design of visual servoing. As the visual sensor delivers the feedback signal to the controller which drives the manipulator with desired motions, the latency in the feedback loop consists mainly of four components: image acquisition delay, image processing delay, delay caused by determining the control signal, and delay for transmitting signal to the manipulator. The latency is governed by image processing in most vision-based control system. For example, with the development of high-speed cameras, the delay for capturing images is largely reduced, e.g. a camera running at a frame rate of 1000 Hz can capture images with a sampling interval of only 1 ms. The long image processing delay slows down the sampling rate of the visual feedback, degrades the control performance, and may cause the system to be unstable. Therefore, the feedback time delay challenges the control design for vision-based control systems [113].

One approach to ensure the stability of the visual servoing systems is to derive the upper limit of acceptable delay. For example, in [114] the delay boundary is determined by using the Nyquist Stability Criterion. A delay beyond the limitation leads to a unstable system. Another approach is to compensate the feedback delay introduced in the feedback loop, e.g. using an extended Kalman filter [115] for pose estimation in [116], and a generalized predictive controller (GPC) [117] considering model dynamics in [118]. For objects with constant motion, e.g. objects on a convert belt or on a moving train, the prediction algorithm gives reliable estimation for manipulator control. However, if the target motion is unmodeled, e.g. objects move randomly in the operating space, the performance of the prediction becomes poor.

Moreover, adopting the shared networks in the control loops presents additional design problems such as bandwidth constraints, delays and packet dropouts. Utilizing the computation power over the communication network increases the computational capability. However, communicating data over the network, such as transmitting image data to remote processing nodes, suffers from additional transmission time delay. In Ethernet, transmission time delays are generally random. During image processing the computation delay is also random due to data-dependent conditional ranches/loops [119]. For systems with random delays various control approaches have been proposed in the literature. In the past, networked control system (NCS) research, the networked-induced delays have been considered, see [120, 121]. In this thesis, the focus is on NVSC system with random computation and communication delays as well as non-equidistant sampling intervals.

Approximating a random delay by its upper bound, e.g. the worst case, results in a robust but conservative control design [122]. Potentially available stochastic models in terms of probability distributions of the delay are discarded. Less conservative control design approaches are based on stochastic analysis [123–132], see [133] for a general overview. In [123], the random delay is considered as an independent and identically distributed (i.i.d.) binary random process. The associated stability conditions and controller design algorithms are derived using the statistical properties of the delays. A Markov process is used to model the random delays as a stochastic process in [124–126]. The resulting closed loop system is Markovian jump system (MJS) with stability conditions and controller design algorithms being determined either by the infinitesimal generator of the delays in continuous-time modeling or by the transition matrix of the delays in discrete-time modeling. However, a constant sampling interval is assumed in most of the existing results, which cannot be applied to the NVSC system with time-varying sampling intervals.

Performance and Cost Issues

It is desirable to improve the control performance with high-speed visual feedback by means of distributed computation over the network. However, the transmission of large volume image data results in a high communication network load, which leads to large communication delays and high packet drop rates. A communication network load beyond the channel capacity may even result in a complete communication blackout. The communication constraints are critical in particular in wireless communication networks. Therefore, in the presence of limited communication bandwidth keeping the data transmission over the communication network at a minimum under the constraint of sufficient control performance is necessary. A scheduler is demanded to cope with the communication constraints in particular in the case of large volume image data. A general overview about the related works on the co-design of control systems and communication networks can be found in [121, 134–136]. Some of the important results are summarized in the following.

In order to study the problem of optimal control and network scheduling over limited bandwidth deterministic networks, a set of linear time-discrete systems are modeled in [137]. The optimal control and scheduling problem is formulated into mixed integer quadratic programming formulation, and the results are applied in the simulation of a car suspension scenario. An adaptive controller, which adjusts its output according to network delays, is proposed considering the quality-of-control in NCSs in [138]. Dynamic management of the network traffic is realized through the control adaptive techniques. In [139] a predictive control and scheduling co-design approach is proposed for a set of networked control systems. The system stability is guaranteed by ensuring the time delay within an analytical upper bound. A predictive controller is designed by compensating the delay with the delayed sensing data and the previous control information. However, this approach, which requires an exact knowledge of systems and the delays, is susceptible to the modeling uncertainties and the non-deterministic network characteristics. In order to reduce the communication network traffic, the deadband control approach is proposed in [140, 141], which only sends packets when the current value exceeds a given threshold. The benefits of such an event-triggered sampling scheme over a periodic sampling scheme are analytically shown for the first time in [142]. The optimal combination of control and event-triggered sensor transmission is investigated in [143–145].

Different from the framework of deterministic design approaches introduced above, the probability distribution of network attributes is also considered in system design in the literature. Modeling the network time delay and packet dropouts with stochastic processes such as Markov processes results in control systems and communication co-design with stochastic analysis methodologies [146–150]. The network induced problems such as time delays, packet dropouts, and disorder of data-packet transmission are considered in control and communication co-design in [146]. In order to design an optimal controller, the optimal sampling frequency is determined based on the system dynamics and the network characteristics. However, the transmission delay is assumed to be less than one sampling interval. The proposed approach is not suitable for systems with longer time delays. System analysis for nonlinear NCS employing Ethernet and Ethernet-like wireless and wireline is carried out in [148]. Stochastic scheduling protocols are developed by means of input-output stability analysis. The problem of allocating the communication resources for optimal performance of linear NCSs is studied in [150]. Networks with limited communication rates are modeled by uniform quantization with white-noise quantization errors. The optimal control performance is stabilized by jointly allocating the communication resources and tuning the controller parameters.

2.4 Insect-Inspired Vision for Robotics

A fly's panoramic vision system comprises at its front end several thousands photoreceptor feeding into a 2D array of motion detecting neurons which the animal uses for dynamic visuomotor pose and gaze stabilization, and navigation in six degrees of freedom. The Reichardt model [151] is a well-known model which describes, at an algorithm level, the process of local motion processing in the fly, leading from a non-direction input to a direc-

Table 2.1: Comparison of different devices for EMD implementation

	FPAA	μC	VLSI	FPGA
EMD Array size	−	+	++	++
Energy consumption	−	+	++	+
Design time	+	+	−	+
Cost	+	++	−	+

tion selective output. In a structure of the fly brain called 'lobula plate' large neurons are found which integrates these local motion signals and additionally form extensive connections amongst themselves [15, 16]. These neurons have large receptive fields and respond best to particular low-fields such as occurring during certain maneuvers of the fly in free flight [152, 153].

In order to upgrade the Reichardt model to be an reliable biological elementary detection model, additional components, such as spatial and temporal high pass and low pass filters, thresholding, dynamic saturation and integration are added [154–156]. With the advances in the research of Reichardt model, a number of EMD-based implementations are carried out in the literature [157–162]. Franceschini designed an EMD circuit for a micro-air vehicles (MAV) by using Field Programmable Analog Array (FPAA) and μC (the Cygnal 8051 F300) [157]. However, the FPAA based EMD circuit turns out to be very energy-consuming. It is therefore replaced by a tiny 3×3 mm μC which computed optic flow with relatively small device weighing (only 0.1 g) [158]. In [159–161], EMD based visual processing were carried out on Very-Large-Scale-Integrated (VLSI) circuits which also provide sufficient computation resources. In [162], Harrison has designed and tested a single-chip analog VLSI sensor that could detect imminent collisions by measuring the radially expanding optic flow. An EMD circuit was developed to measure the optic flow of a CMOS sensor, while a 16×16 array of two-dimensional motion detectors was designed for collision detection 100-400 ms ahead of the impact in real-world scene.

In addition, FPGA devices with programmable logic components are also selected for EMD implementation [163–166]. For example, in [163] a motion estimation algorithm with configurable parameters for EMD was implemented on a FPGA board. The proposed algorithm, implemented using the VHDL (VHSIC hardware description language) language in the FPGA device, searches the optimal EMD parameters and computes the global motion vectors in real time. Besides, a FPGA implementation of a insect-inspired visual sensor is introduced in [164], which is designed to estimate the optic flow onboard for Micro-Air Vehicles (MAV). With the advantage of parallel processing capability of FPGA, the signals from up to 245 EMDs could be implemented at a 100 Mhz clock frequency on a $<$ 1 gram piece of integrated digital electronics that requires only a few external components. The specific EMD architecture is integrated into a Virtex2 Xilinx FPGA (XC@V250) $12 \times 12 mm$ in a size of 250,000 system gates.

The implementations of EMD on various devices introduced above have their own advantages and disadvantages as illustrated in Table 2.1. For example, using the FPAA and μC can realize a limited number of EMDs easily and quickly, but has the restriction of large energy consumption and weights. FPGA is a compromise of time for design process, system mass, and flexibility. FPGA has in particular an advantage of having parallel and pipelining ability in each unit. With these remarkable characteristics of FPGAs, it is flexible to design a multi-EMDs system. More specific instructions of the fly's vision system and the EMD variants are given and discussed in Chapter 5.

2.5 Summary

To sum up, high-speed vision based on the networked computational resources or the insect-inspired motion detector improves the sampling rate of the visual feedback, which is critical to system performance and system stability. Until now, there are few works in the literature dealing with either networked high-speed visual servo control or high-performance insect-inspired visual servo control. For NVSC, most works only deal with a subset of the different aspects of visual servoing or networked control. To the best knowledge of the author, almost no work has been done towards networked high-speed vision for visual servo control that concerns the real-time distributed computation, stability analysis, network resource and control performance at the same time. In the field of EMD-based robotics, most state-of-the-art works have used EMD for qualitative motion estimation, yet, only few of those have coped with the quantitative motion estimation shown in simulated behaviours, which, however, limits its expansion of applications to real-world experiments.

For the first time, high-speed NVSC and insect-inspired visual servo control are comprehensively investigated and studied in this thesis. The objective of this thesis is to deal with the aspects of high-speed vision for robot control realized through these two approaches, including real-time image data transmission, stability analysis, data scheduling for optimal cost-performance trade-off, elaborated EMD design and EMD-based motion estimation. The presented work contributes to the state-of-the-art in the domains of data transmission, networked control, vision-based control, and insect-inspired robotics.

3 Networked High-Speed Vision for Visual Servoing

One important task of vision-based control systems is objects localization. The general image processing algorithm for object localization has mainly three steps, namely image acquisition, object recognition, and pose estimation. The algorithm for object recognition consists of image feature extraction and image feature matching. Since extracting robust and discriminative features from image sequences is the most critical step and yet, the most difficult step, different approaches and methodologies have been developed in the past decades. The selection of an appropriate operator for feature extraction is dependent on different applications. Some algorithms are developed to extract low-level image features such as edge, corner, and optical flow, which contains no shape information [167]. As these basic operators are sensitive to the changes in image scale and orientation, they are not suitable for the applications, where feature matching is needed after the step of feature extraction. Therefore, more advanced feature extraction algorithms such as the scale invariant feature transform (SIFT) [9, 109] are developed in order to improve the performance of feature detection and feature matching in object recognition. The features detected in the manner of SIFT are less sensitive to image scale, image rotation and the varying illumination compared to the features obtained by using basic operators. However, the computation delay by applying the advanced algorithms also increases. Apparently, there is a compromise between computational cost and efficiency in selecting image feature operators. For a visual servo control system, the choice of advanced feature extraction algorithms benefits the image processing results, and introduces, however, challenges to system analysis and controller design regarding the long computational time.

In order to reduce the computation time caused by image processing, advanced hardware and simplified algorithms are adopted in some applications. On one side, specific hardware such as FPGA and GPU are utilized to increase the computation capability of parallel processing. On the other side, advanced algorithms with guaranteed performance, e.g. using SURF (Speed Up Robust Features) [168] to replace SIFT, are favored by some works. However, the resulting sampling interval of the visual data is usually still longer than 30 ms, which largely exceeds the cycle time of a robot joint position loop. For example, it takes about 20 ms to extract SURF (Speed Up Robust Features) [168] from an image of a size 640×480 pixels on a graphic card NVIDIA-GeForce 8800, while the sampling interval of the joint position control loop is typically in the scale of 1 ms. For highly dynamic vision-based motion control, e.g. tracking of fast-moving objects, the long image processing delay increases the risk of losing the objects in the field of view, and thus, may cause control failures.

In order to overcome the low sampling rate problem of the visual feedback, utilizing computation resources over the network to speed up visual feedback becomes an alternative as to the methods discussed previously, see an example in Fig. 3.1. Instead of centralized computation, distributed image processing is carried on several processing nodes. Each processing node receives a subset of the original image sequence over a communication

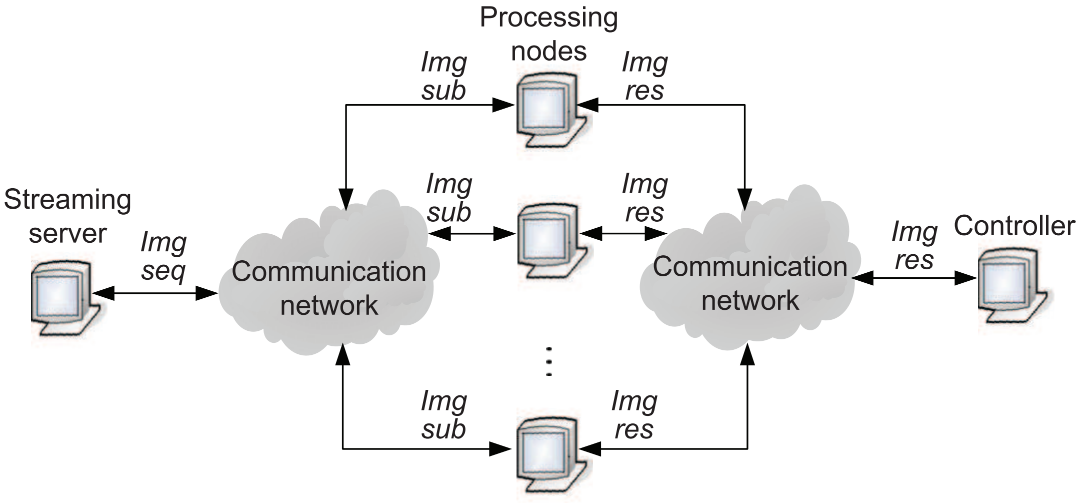


Figure 3.1: Distributed computation platform with ncRTP for parallel image processing in networked visual servo control systems, consisting of image streaming server, controller, and processing nodes. *Img seq*: original image sequence captured by camera; *Img sub*: subset of the original image sequence; *Img res*: image processing results.

network from a streaming server. The obtained image processing results are then sent to the controller to calculate the control signal for the manipulators.

The goal of this chapter is to increase the sampling rate of the visual feedback and the robustness of the perception results for closed loop control. Contributions of the presented work are a promising cloud image processing platform and a robust image processing algorithm for vision-based control. Issues towards the investigation and the design of real-time transport protocol for large volume image data transmission over the communication network are addressed. For the design of robust image processing algorithm, it requires a systematic analysis of the scale space in feature detection and feature matching.

The remainder of this chapter is organized as follows: A high-speed vision system with real-time image transmission protocol is developed in Section 3.1. In Section 3.2 the analytical design of binomial distributed sampling in scale space and the self-scaling mechanism is introduced to improve the performance of image processing. The characteristics of the cloud computing platform, in particular the property of the feedback delay, are discussed in Section 3.3.

3.1 High-Speed Vision over Network

In order to accelerate visual feedback, it is desirable to perform image processing in a parallel way. For example, distributing images to different processing threads/nodes can increase the frequency of image processing. Instead of using a super-power computer, the computation resources connected over a common network are considered for parallel image processing. Since the most time consuming part of the image processing is the algorithm for feature extraction from the original images, it is considered to transmit the raw image data over the communication network in this work. A whole gray-level image, rather than sub-blocks of the image, is transmitted to a single processing node for feature extraction. The reason is that a sub-block of the original image loses the information of the adjacent pixels on the margin. Accordingly, the feature operator running on the sub-blocks gives different results from that on the original image. Moreover, distributing sub-blocks to different processing nodes requires additional inter-frame synchronization mechanism, resulting in a growth in the complexity of transport protocol design. Therefore, it is beneficial to transmitting whole image frames over the communication network.

The structure of the proposed high-speed vision system is shown in Fig. 3.1. In this structure, a single-sensor system is formulated with an emphasis on image data transmission and parallel computation. Expanding the single sensor system in Fig. 3.1 to a multi-sensor system will lead to other technical issues, such as synchronization and data fusion, which are outside the scope of this thesis.

3.1.1 Real-Time Image Transmission

A single channel image with a resolution of 640 × 480 pixels possesses a size of about 300 KB. For a camera running at a frame rate of 60 Hz, it requires a bandwidth of about 144 Mbps. Transmitting these massive image data over the communication network becomes a challenging task in NVSC systems.

In order to accomplish this task, an appropriate device has to be chosen to build the physical layer. As discussed in Section 2.2, the 1000BASE-T Ethernet is implemented to transmit image data, as other data link/physical layer devices such as EtherCAT, CAN, SERCOS or WLAN either need dedicated hardware or have limited transmission rates. Besides, an image transmission protocol *networked control real-time transport protocol* (ncRTP) based on real-time transport protocol (RTP) is developed for NVSC systems. Although there are various transport protocols in the public domain, they are not suitable for real-time image transmission[1]. The RTP supports the delivery of data with real-time characteristics and provides the services, including payload type identification, sequence numbering, timestamping, and delivery monitoring. The developed transport protocol ncRTP with the 1000BASE-T Ethernet as the physical layer is briefly introduced in this section.

Network Stack

In order to design the transport protocol for image transmission over the network, the network stack is briefly introduced here. The computer network is a packet-switched telecommunication network. Data are encapsulated in different units depending on the physical devices, such as Ethernet, ATM, PPP and so on [169]. Higher level layers are developed to support communications independent on the devices. Each layer provides different services by adding new functions. Meanwhile, the new layer depends on the services of the underlying layer. The structure of layers 2-5 in the OSI (open system interconnection) model [169] is illustrated in Fig. 3.2. From this model, a suitable layer is selected for image transmission. The features of each layer are described below:

- Layer 2 (Data Link): *Physical addressing.* This layer is device dependent and the data unit in this layer is called frame existing only in LANs.

[1] See discussion in Section 2.2

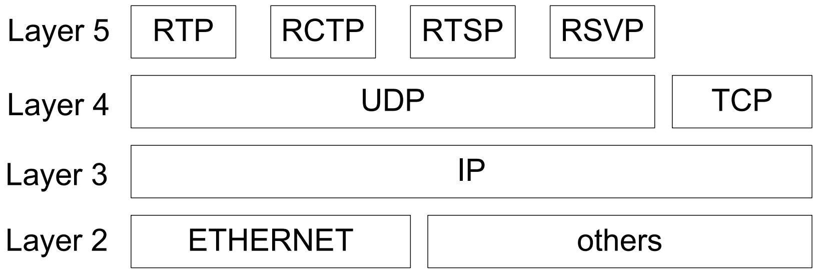


Figure 3.2: Layers 2-5 in the OSI model [169].

- Layer 3 (Network): *Path determination and logical addressing.* This layer only provides essential functions for data transmission crossing LANs.
- Layer 4 (Transport): *End-to-end connections.* It is an IP address multiplexer. UDP provides basic functions, while TCP implements extra features, e.g. reliable transmission, error detection, flow control and congestion control.
- Layer 5 (Application): All functions not mentioned in the underlying layer could be implemented here.

According to layer 4, UDP and TCP are commonly used protocols on the Internet. However, they are not suitable for real-time transmission tasks, which require large volume data exchanges. UDP, commonly used for streaming audio and video, has no form of flow control or error correction. Thus, it cannot be used to send important data. Besides, it is not able to transmit data package larger than 64 KB [169]. Different from UDP, TCP offers error correction and provides guaranteed delivery. However, TCP has a larger absolute value and also larger variance of the transmission delay due to windowing behavior and retransmission mechanism. A solution based on both UDP and TCP is proposed in this thesis as a compromise between the speed and the performance.

A new protocol is designed which combines the features of UDP and TCP. The new protocol is not necessarily designed from the draft. It is based on the RTP [170], which is an application layer protocol for transmitting latency-sensitive data, such as video and audio on the Internet. It is the foundation of many Voice over IP and media streaming systems. The RTP uses a separate channel for monitoring and adjusting the data delivery. In this thesis, ncRTP is designed and implemented based on GNU ccRTP [171], which supports both UDP and TCP as transport layer protocol. Since TCP is unsuitable for real-time applications, UDP is used as the underlying protocol for ncRTP to build a low-latency protocol stack providing multicast support. As a support layer protocol, UDP inherits features of IP protocol, which is characterized as connectionless and unreliable. In comparison with TCP, UDP has no transport level mechanisms to ensure the data integrity such as flow control and retransmissions. These mechanisms can improve the transmission reliability with, however, the payout of extra unpredictable latency. In order to obtain a balance between the robustness and the performance, some TCP features are implemented on the application layer in ncRTP. The features of ncRTP are described in the following.

RTP Header

The RTP has a header layout shown in Fig. 3.3 (refer to [170] for details of the meaning of different fields). It has to be mentioned that PT denotes the payload type, e.g. the

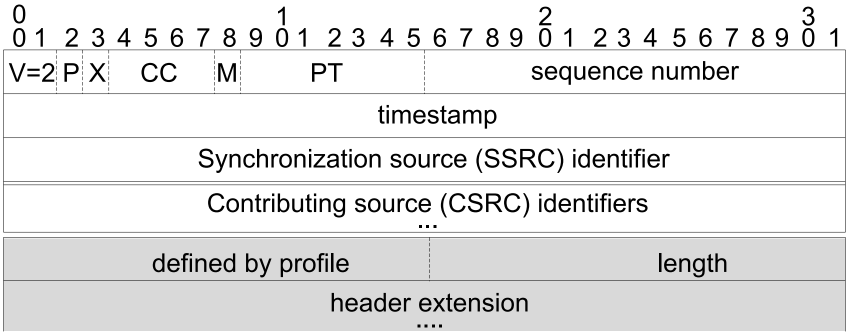


Figure 3.3: RTP fixed and extended header fields (modified from [170]). Upper part: fixed header; lower part: extended header. V: version=2; P: padding; X: extension; CC: CSRC count; M: marker; PT: payload type.

different image types. The sequence number is assigned to each RTP data packet which can be used by the receiver for both packet restoring and packet loss detection. Besides, the 32-bits timestamp that reflects the sampling instant of the first octet is utilized as the index for the detection of packet disorder at the receiver side. The header extension here allows to carry additional information required for individual implementations, e.g. the image frame number and the timestamp when the image is captured.

Fragmentation Mechanism

For transmission of data larger than 64 KB on top of UDP protocol, such as single-channel image (640 × 480 pixels) with a typical size of 300 KB, the data have to be fragmented by the sending module and assembled by the receiving counterpart. However, the fragmentation mechanism is not defined uniformly for different payload types in ccRTP. Here, a transparent data fragmentation/reassembling layer is built in ncRTP in order to support diverse data formats from different cameras in NVSC applications. Large frame is truncated as a series of data blocks patched with an extended identification number. For example, an image with a size of about 300 KB is segmented into five data blocks with each block smaller than 64 KB, see Fig. 3.4 for visualization. Then, the receiving stack can detect and reassemble the frame according to the identification number and the extended payload types defined in the header of ncRTP.

Congestion Control

Most packet loss in Internet is caused by congestion [172], which could lead to congestion collapse. The system time delay increases continuously if the data are lost successively. The sliding window mechanism [173] of TCP, by which the transmission stream is expended to consume all available bandwidth, is not selected as the congestion control strategy for ncRTP. In contrast, the data are transmitted with a certain rate by RTP. Several methods have been proposed for RTP congestion control in [174]. In ncRTP the congestion control is implemented as image resolution/rate adaptation based on control protocol (RTCP) feedback. If there is packet loss, the whole application data unit (ADU), e.g. a frame, is dropped. Instead of retransmitting this frame, the next frame which is available and contains more important information for networked control will be sent in this case. When

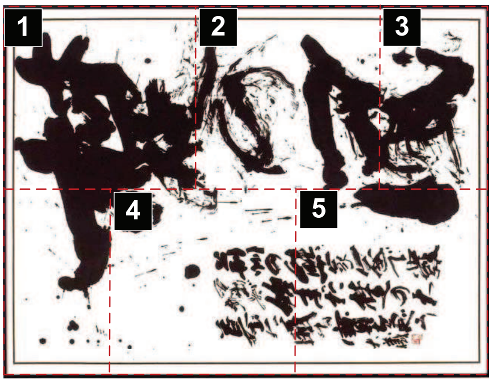


Figure 3.4: Image data fragmentation. The image with a size of 640 × 480 pixels is segmented into five data blocks. The size of each block is smaller than 64 KB.

successive data losses are detected by the receiver, the sender could be notified through RTCP to decrease the sending rate or to reduce the image resolution.

Synchronization and Timing

The distributed computation system is designed as an event-driven system. The data processing thread is triggered when a new frame has been assembled in the receiving buffer. Therefore, the system works with the same frequency of the grabbing rate of cameras. The real-time clock (RTC) of the hosts in NVSC systems can be synchronized by network time protocol (NTP) up to 0.1 ms. The RTP header is extended according to [170] and the RTC timestamp is appended in the extension. The streaming server is designed on RTAI real-time kernel and the sending process is scheduled periodically within a jitter of 10 μs. Based on this framework, the control performance could be evaluated with small granularity.

System Parameter Calibration

The implementation of the ncRTP stack requires a large throughput capacity to transmit image streams with high frame rate. In order to avoid overflow on the receiver side, a double-layer cascade buffer structure is adopted. The UDP packets are removed immediately from the first buffer to the second one if an image frame has been reassembled successfully. The parameters of the operating system such as Ubuntu Linux have to be optimized for general-purpose applications. They should be calibrated for large volume data transmission to avoid packet loss. In the experiments, the parameters such as buffer size of IP stack and UDP packet size need to be tuned for stable transmission.

With the transport protocol introduced above, efficient transmission of large volume image data over the communication network is realized. The transport protocol ncRTP can be installed on PCs where image streaming or image processing is carried out.

Remark 1 In this work, ncRTP is applied only in one-sensor systems. Expending it to multi-sensor systems, or even more complex multi-sensor multi-streaming-server systems, requires the design of additional mechanisms. For example, an additional pre-processing module for the coordination of the source data from different sensors belongs to part of future work.

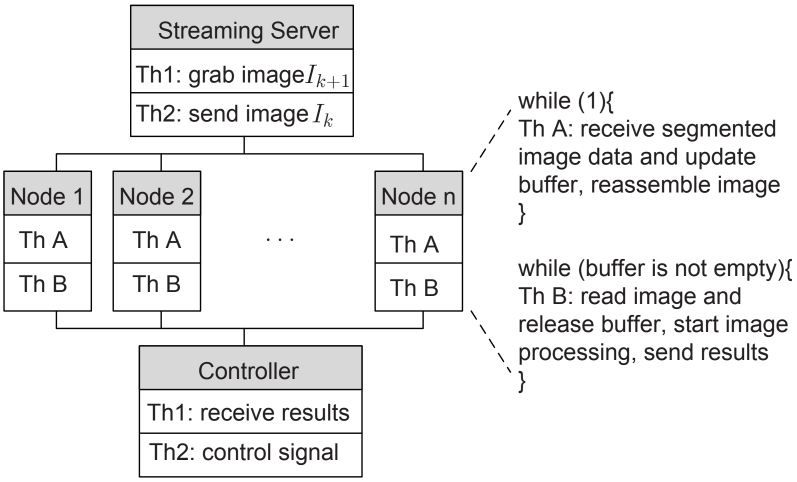

Streaming Server
Th1: grab imageI_{k+1}
Th2: send image I_k
Node 1
Th A
Th B
Node 2
Th A
Th B
. . .
Node n
Th A
Th B
while (1){
Th A: receive segmented image data and update buffer, reassemble image
}
while (buffer is not empty){
Th B: read image and release buffer, start image processing, send results
}
Controller
Th1: receive results
Th2: control signal

Figure 3.5: Thread scheduling on streaming server, processing nodes and controller in NVSC systems. I_k is the image captured at time instant t_k.

3.1.2 Cloud Image Processing

Based on the ncRTP introduced above, a cloud image processing platform shown in Fig. 3.1 is established to achieve high-speed visual feedback. The cloud image processing platform, which utilizes existing computing resources, is more flexible and economical compared to the conventional approaches. This platform consists of a streaming server, several processing nodes, and a controller. In order to further reduce the computation time for image processing on each single processing node, an GPGPU implementation is applied by exploiting its massive parallel computation capacity developed in recent years.

The features of the streaming server, the processing nodes and the controller in the cloud image processing platform shown in Fig. 3.1 are introduced below.

Streaming server

The streaming server is equipped with a high-speed camera capturing images at high frame rate. The captured images are sent through the sending process scheduled by the real-time RTAI kernel. In order to reduce the latency, two real-time tasks run in parallel as shown in Fig. 3.5. One task polls periodically the image from the camera frame buffer, which is necessary especially for cameras not supporting external trigger mode. The other one is triggered when the first task has received a whole frame from the camera. Then, the image data is sent immediately together with a frame index through ncRTP. Multi-streaming is also supported by ncRTP. In order to reduce the network load, sending rate scheduling algorithm considering control error and/or network status could also be implemented on the streaming server.

Processing nodes

The ncRTP protocol is installed on the nodes and is optimized for receiving large volume image data. Moreover, fast, accurate and robust image processing is required with respect to a good control performance, in particular for highly dynamic vision-based motion control. The GPGPU implementation of image processing algorithm is adopted in this work.

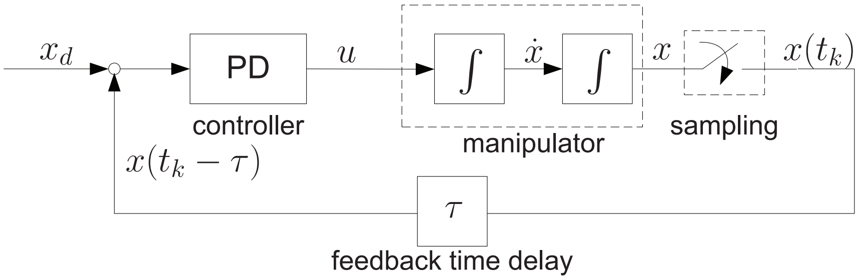


Figure 3.6: System diagram of a system for position control in the simulation. x_d: desired position; x, $\dot{x}$: current position and velocity of the manipulator; t_k: sampling instant. The feedback time delay is bounded by $15\,\text{ms} \le \tau \le 35\,\text{ms}$.

Each processing node within this platform is equipped with a graphic card. For position-based visual servoing (PBVS) considered in this work, image processing algorithm consists of feature extraction, feature matching, outliers rejection, and pose estimation. The output of image processing gives the relative pose (including the translation and the rotation) between the object and the camera. The image processing algorithm is scheduled as soon as an image has been reassembled in the receiving buffer.

Controller

The controller could be implemented on another node or on the same one used as streaming server or image processing node. It receives image processing results (< 64 KB) as well as the frame index from the processing nodes through UDP. At the controller side, the received data are pre-filtered by sorting it along the frame index. The data with smaller frame index as that of the latest received data will be dropped. Thus, the packet disorder problem is prevented. Finally, the image processing results are used to determine the configuration signal for the manipulator to realize the control task.

With the platform introduced above, a parallel image processing architecture is established, and thus high-speed visual feedback is achievable for NVSC systems. Here, in order to validate the advantage of visual feedback with high sampling rate, a simulation of position control on a 1-DoF manipulator is conducted. A PD controller is applied. The system diagram is shown in Fig. 3.6. A double integrator plant is considered. A stochastic time-varying feedback delay bounded by [15ms, 35ms] is simulated. Sampling rates of $\{40\,\text{Hz}, 80\,\text{Hz}, 120\,\text{Hz}, 200\,\text{Hz}\}$ are applied for tracking control. The PD parameters are chosen heuristically. The number of available processing nodes is denoted as g. If one node deals with image processing at a frequency of 40 Hz, with g processing nodes running parallel as shown in Fig. 3.1, it results in an image processing rate of $g \times 40\,\text{Hz}$. The simulation results are shown in Fig. 3.7. It is observed that, with a higher sampling rate the control error becomes smaller, e.g. the mean control error of 80 Hz is about 28% smaller than that of 40 Hz, see Fig. 3.7 (b). Based on the results, it is concluded that it is beneficial to utilize the cloud image processing platform for high-performance visual servo control.

Another important issue of NVSC systems is to obtain reliable image processing results. The design of robust image feature extraction and feature matching is introduced in the following section.

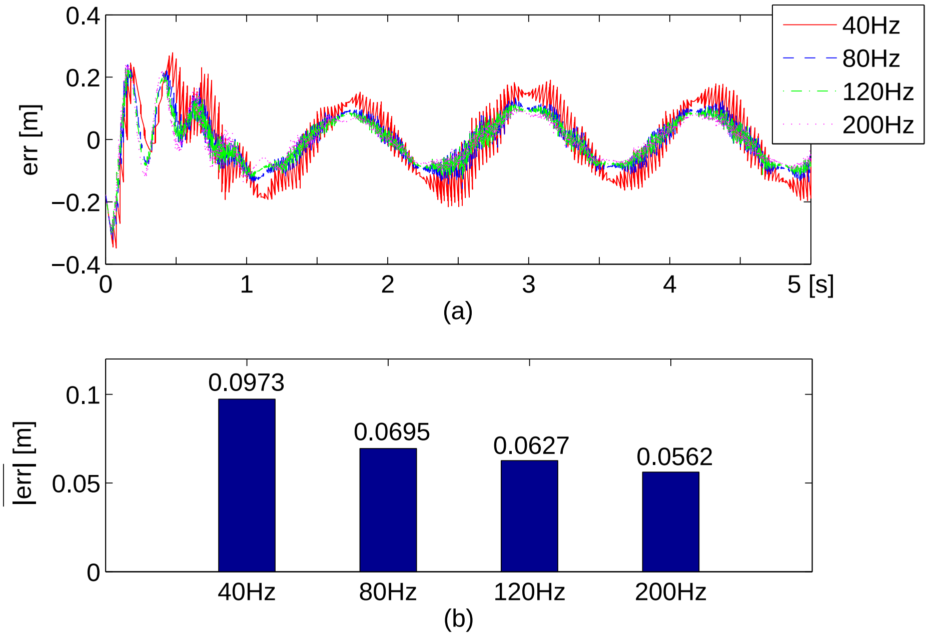


Figure 3.7: Simulation results of control performance versus sampling rate of vision signal. (a): Tracking error; (b): Mean tracking error. Desired trajectory: $\sin(t) + \sin(5t + \frac{\pi}{5}) + 0.7\sin(2t + \frac{\pi}{6})$; feedback time delay: [15ms, 35ms]. Sampling rates $\{40, 80, 120, 200\}$ Hz are tested.

3.2 Robust Image Processing

With the cloud computing platform developed in the proceeding section, parallel image processing is achievable. Accelerating image processing on individual processing node could reduce the sampling interval of the visual data, and thus the feedback delay. In this section, it is concerned with improving the accuracy of image processing results with a focus on the performance of image processing on each processing nodes. For high-performance vision-based control the image processing step is essential since it provides the necessary perception results for the following control step. Error in image processing results gives wrong information about the surroundings, and may degrade the control performance.

In this work, a 3D object tracking problem is considered. The objective of image processing is the estimation of the position and the orientation of the object based on the images captured by a monocular camera. Feature extraction, feature matching, and calculation of the camera extrinsic parameters are the three main steps in the estimation algorithm. This section aims at improving the performance of 3D pose estimation by focusing on the analysis of feature extraction and feature matching algorithms. In the rest of this section, a two-step method is proposed to extract features considering the scale of the object in the image. A binomial sampling in scale space is applied for feature detection. Then, a self-scaling mechanism is proposed to improve the feature matching rate as well as its accuracy. The performance of the proposed approach is evaluated in the experiments with comparison to the standard approach.

3.2.1 Sampling Optimization in Scale Space

The goal of 3D pose estimation for position-based visual servoing is to identify the translation and the rotation of an object. For high-performance tracking, the estimation algorithm is required to be robust to certain translation, rotation, scaling, and small distortion of the images. To this end, some algorithms based on scale space theory have been developed in recent years such as SIFT [9] and SURF [168]. Considering the high performance achieved by image processing based on scale space theory, feature detection and feature matching in scale space are analyzed and extended for robust image processing. The image structural analysis in scale space is introduced below.

Scale Space Theory

The Laplacian operator measures the second spatial derivative of an image, and is often used for edge detection and blob detection. The Laplacian $\bigtriangledown^2 I$ of an image with image intensity $I(x, y)$ at the pixel (x, y) is given by

$$\bigtriangledown^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2},$$

which can be achieved by using a convolution filter. However, this operator is very sensitive to noise. Therefore, the image has to be smoothed with e.g. Gaussian smoothing filter before applying the Laplacian operator. The Gaussian kernel with standard deviation σ is given by

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}.$$

Due to the associativity of the convolution operation, the Gaussian smoothing filter is convolved firstly with the Laplacian filter, which yields the Laplacian of Gaussian (LoG) kernel

$$\bigtriangledown^2 G(x, y, \sigma) = -\frac{1}{2\pi\sigma^4}(2 - \frac{x^2 + y^2}{\sigma^2})e^{-\frac{x^2+y^2}{2\sigma^2}}.$$

Selecting the value of the standard deviation σ depends on the scale of the object. Since the objects in real-world have different scales, and depending on the scale of observation they may appear in different ways, it is motivated to generate a scale space representation. The Gaussian kernel is a canonical choice for generating a scale space representation [175–177]. The variance of the Gaussian kernel σ is referred to as the *scale parameter*. In other words, the scale space family is the LoG filter responses of different scales. Since using the LoG kernel for feature extraction results in high computation cost, different methods are designed to approximate LoG with reduced computational complexity. For example, the Difference of Gaussian (DoG) is used in SIFT [9], while the Determinant of Hessian (DoH) is applied in SURF [103].

Binomial Distributed Sampling in Scale Space

With features extracted by the approximation of LoG in different scales, the next step is to perform feature matching. The features detected in the current image are compared with that detected in the reference image through the feature matching algorithm such as the nearest neighbor technique [178]. An example of the matching results based on SURF is shown in Fig. 3.8. The performance of feature matching is evaluated with different

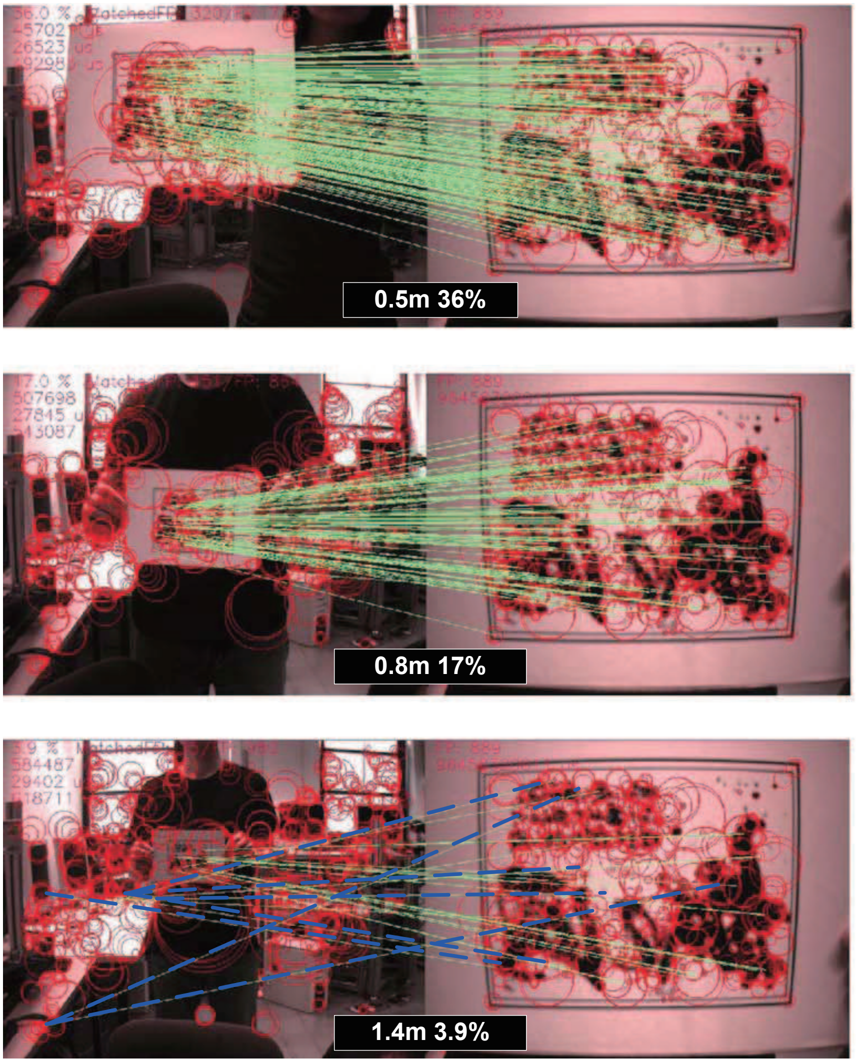


Figure 3.8: Feature matching results based on SURF. The center of the circle denotes the location of feature point in the image plane, while the radius denotes the scale of the feature. The distance of the object to the camera is: 0.5 m (top-left), 0.8 m (middle-left), and 1.4 m (bottom-left). The number of features is: 718, 864, and 980 respectively. Right column: reference image. The matching rate is: 36%, 17%, and 3.9% respectively. The dashed lines in the bottom frame denote incorrect matching pairs.

object distance: 0.5 m, 0.8 m, and 1.4 m. The image processing results are summarized in Tab. 3.1. It is shown that the matching performance degrades rapidly with increasing distance. With the object distance of 1.4 m the matching rate is only 3.9%. Besides, an increase of the object distance also causes error of feature matching (see dashed lines in the bottom panel of Fig. 3.8), which is intolerable for the following pose estimation algorithm.

The main reason for the low matching rate and the high matching error is that the sampling in the scale space is not well distributed. The optimization for feature detection and feature matching on the scale space level is less considered in known literature. In this part the sampling problem in scale space is analyzed and a matching approach with improved performance is proposed.

Table 3.1: Feature matching results based on SURF.

Distance [m]	Feature number	Matching rate
0.5	980	36%
0.8	814	17%
1.4	718	3.9%

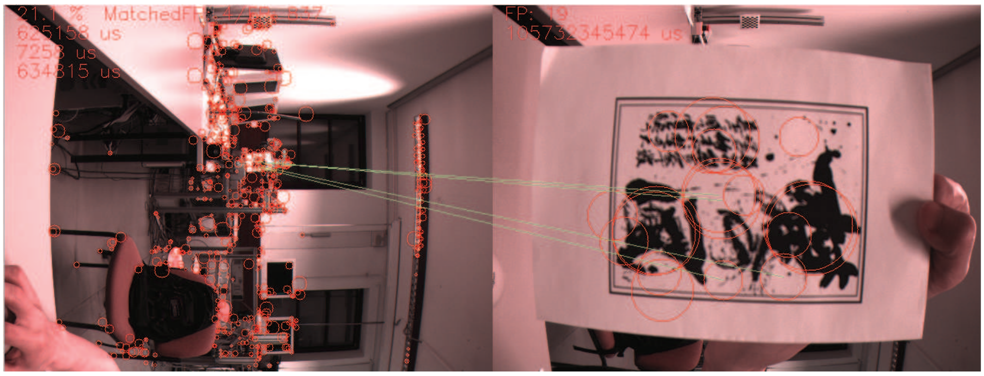


Figure 3.9: Object in images with different scales. The details of the object disappear rapidly when the object is further away. The size of the object in the right/left image is $320 \times 240\,/16 \times 12$ pixels.

Simulation of the Image Information Diffusion The scale space describes the image information diffusion. Assume an object moving away along the direction of the camera's optical axis. The projection of the object on CCD diffuses, see Fig 3.9 for example. On the right hand side of Fig. 3.9 the size of the object in the image is about 320×240 pixels, while on the left hand side the size of the object is only 16×12 pixels. The detailed information of the object disappears gradually when moving the object away from the camera. In other words, if there is a scale space simulating the diffusion process precisely, the matching of the object in the current image to that in the reference can be performed with high accuracy.

In order to simulate the diffusion process, a multi-resolution scale space is suggested, see Fig. 3.10 for an example. This scale space has five octaves, and each octave contains three scales with increasing scale parameter σ. In each octave the images have the same resolution. The image resolution of the next octave is smaller than that of the previous octave. With digital image processing, only the sub-sampling with a factor 2 can be calculated directly with low computation cost. Therefore, the established scale space is discrete and coarse, which is thus difficult to simulate the continuous information diffusion of the object. For high-performance feature matching, it is desirable to design a well-distributed scale space which simulates the information diffusion appropriately.

Two-Step Feature Detection In order to overcome the problem of the ill-distributed sampling due to the sub-sampling with a factor of 2, a novel two-step feature detection approach is proposed in this work. The algorithm starts with rough feature detection, which is followed by a precise feature detection step. The block diagram of the algorithm is shown in Fig. 3.11.

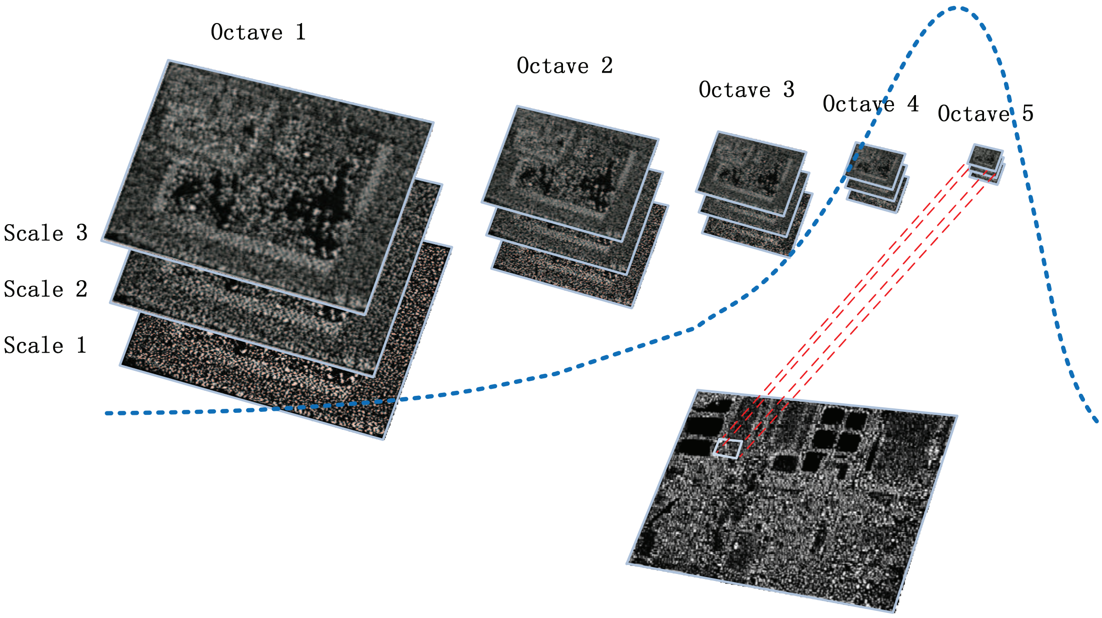


Figure 3.10: Construction of scale space, and binomial distributed sampling in scale space for optimal matching. The scale space has 5 octaves, and each octave has three scales.

The objective of the first step is to determine the scale of the object approximately in the scale space. In order to save the computation time, a scale space with a coarse sampling over the whole scale space is applied. Besides, this sampling is chosen to be evenly-spaced for simulating the object with different scales. Assume the course scale space has I octaves, and each octave has J scales. In order to determine the scale parameter σ, the ratio of the image resolution to the scale parameter σ is considered

$$r_{i,j} = \frac{R_i}{\sigma_{i,j}}, \quad i = 1, \dots, I, \quad j = 1, \dots, J, \tag{3.1}$$

where R_i denotes the resolution of the image in the i-th octave, and $\sigma_{i,j}$ denotes the scale parameter for the j-th scale in the i-th octave. As the resolution of each octave (with subsampling factor of 2) is known, the value of the scale parameter σ for each scale in the same octave is chosen such that the sampling gap between the i-th octave with the resolution of $R_i = \frac{R}{2^{i-1}}$, and the $i+1$-th octave with the resolution of $R_{i+1} = \frac{R}{2^i}$ is evenly fulfilled, where R denotes the resolution of the original image. In general, the scale parameter of the first scale in each octave can be heuristically selected. For example, the scale parameter for the first scale in the octave $i/i+1$ is $\sigma_{i,1}/\sigma_{i+1,1}$, known a priori. Then, the difference of the ratios defined in (3.1) between these two scales is

$$\Delta r_{i,i+1} = r_{i,1} - r_{i+1,1} = \frac{R_i}{\sigma_{i,1}} - \frac{R_i}{\sigma_{i+1,1}} = \frac{R}{2^{i-1}\sigma_{i,1}} - \frac{R}{2^i\sigma_{i+1,1}}.$$

In order to cover the scale space uniformly, the scale parameter $\sigma_{i,j}$ for the scale j in the octave i is selected as

$$\sigma_{i,j} = R_i(r_{i,1} - \frac{j-1}{J}\Delta r_{i,i+1})^{-1}, \quad j = 1, \dots, J,$$

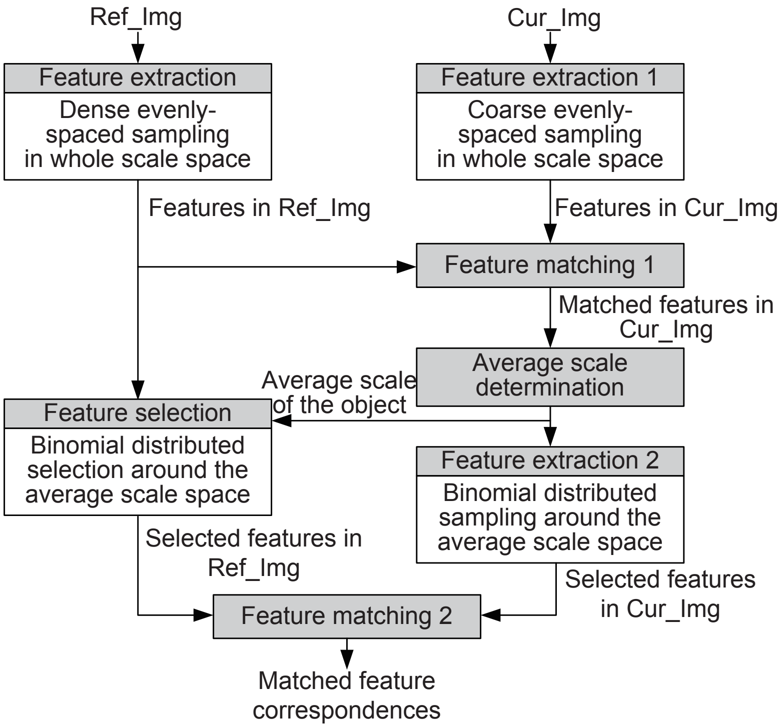

Ref_Img
Cur_Img
Feature extraction
Dense evenly-spaced sampling in whole scale space
Feature extraction 1
Coarse evenly-spaced sampling in whole scale space
Features in Ref_Img
Features in Cur_Img
Feature matching 1
Matched features in Cur_Img
Average scale determination
Average scale of the object
Feature selection
Binomial distributed selection around the average scale space
Feature extraction 2
Binomial distributed sampling around the average scale space
Selected features in Ref_Img
Selected features in Cur_Img
Feature matching 2
Matched feature correspondences

Figure 3.11: Two-step feature detection and feature matching for robust image processing. Ref_Img: reference image; Cur_Img: current image.

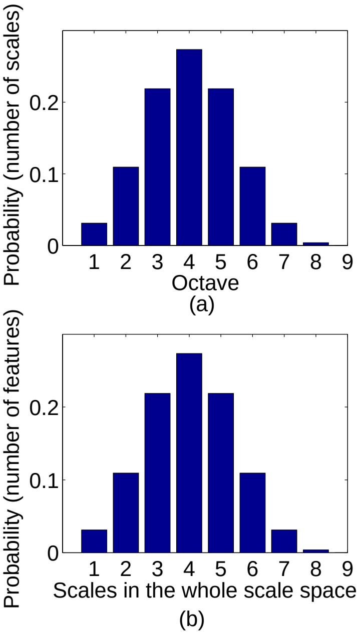


Figure 3.12: Binomial distribution for scale sampling (a) and feature selection (b).

which determines the scale parameter for a scale space with evenly-spaced sampling. Finally, based on the selected scale parameters a coarse scale space of the object from large scale to small scale is generated for feature detection. The coarse scale space is applied to the current image, from which the features are extracted for the feature matching step.

Before performing feature matching, it is necessary to extract features from the reference image. In oder to simulate the information diffusion of the object in different distances, a dense evenly-spaced sampling in the whole space is applied to the reference image. The features of the current image detected in the course scale space are compared with that in the reference image. After matching the features in the current image with that in the reference image, the features on the object in the current image are determined. Then, the average scale of the features on the object is calculated, see Fig. 3.11.

In order to detect more features around the mean scale of the object and to save computation cost by omitting detection in other scales, a second step of precise feature detection is followed for the current image. A binomial distributed sampling is proposed around the average scale, see Fig. 3.10 for an example. Denote the binomial distribution by $K \sim B(n,p)$, where K is a random variable following the binomial distribution with parameters n and p. The probability of being success for k times in n trials is given by

$$\Pr(K = k) = \binom{n}{k} p^k (1-p)^{n-k}, \quad \binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

An example of binomial distribution is shown in Fig. 3.12 with $n = 8$, $p = 0.5$. In order to detect the features around the average scale of the object, the binomial distribution is applied with the x-axis denoting the octave, and the y-axis denoting the number of scale in the corresponding $n-$th octave calculated by

$$l_n = \Pr(K = n) \cdot \mathrm{Max}_s,$$

where Max_s denotes the maximum number of scales in one octave. For example, in Fig. 3.12 (a) the average scale of the object falls in octave 4, and it has the number of scale layers $l_4 = 27.34\% \cdot \mathrm{Max}_s$.

With the number of scale layers in each octave determined above, a binomial sampling in the scale space is established. Then, the feature detection algorithm runs again with the emphasis on the average scale and its adjacent scales. In this case, the object with even long distance to the camera can still be detected with the binomial distribution centered at the small scales.

3.2.2 Self-Scaling Mechanism

With the two-step algorithm introduce above, features in both current and reference images are detected. For the following feature matching step, the matching strategy in most works carries out the matching with the features of the reference image detected in the whole scale space. It is necessary for applications with no scale known a priori. With the optimized sampling in the scale space introduced in the previous section, the scale of the object in the current image can be roughly estimated online with only a small overhead. Based on the scale information, the matching can be carried out only on the best-fit scales, which thus saves computation time, and more importantly reduces the matching error. See Fig. 3.10 for an example, the most stable matching is in the octaves 4 and 5. Therefore, it is necessary to do feature matching centered at the these two octaves.

Here, the binomial distribution around the average scale of the object is applied again to select features from the whole scale space for the reference image, see Fig. 3.11 and Fig. 3.12 (b). The $x-$axis in Fig. 3.12 (b) denotes the scale in the scale space, while $y-$axis denotes the number of features selected from the corresponding scale. Assume there are in total N_f features detected in the reference image. Then, the number of features from the $n-$th scale for feature matching is

$$b_n = \Pr(K = n) \cdot N_f.$$

For example, in Fig. 3.12 (b) for the scale 6 the number of features selected for matching is $b_6 = 10.94\% \cdot N_f$. Through this method, the matching error can be dramatically reduced compared to the matching approach in the whole space.

After selecting the features from the reference image, the nearest neighbor ratio technique is applied for feature matching. Then the correspondences between the feature candidates on two images are obtained. However, the matching results may contain some outliers, which could lead to error in pose estimation. To ensure the accuracy of the pose estimation, the random sample consensus (RANSAC) algorithm [179], which was developed within the computer vision community, is selected for outlier rejection. RANSAC is a general method to estimate the parameters of a mathematical model iteratively from the input dataset. The last step is to calculate the rotation matrix and the translation vector between the two images based on the matching results through pose estimation algorithms, e.g. the linear pose estimation approach [180] and the virtual servo control approach [181].

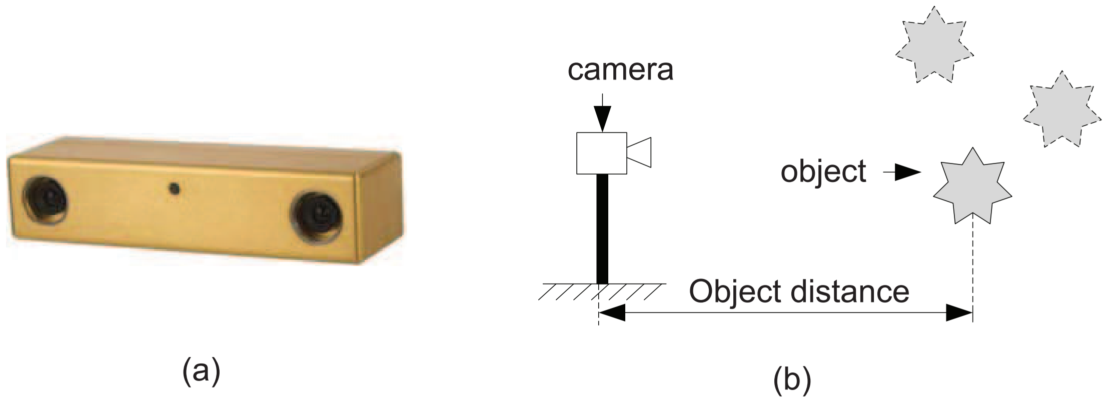


Figure 3.13: Bumblebee®2 camera (a) and the experimental setup (b) for feature detection and feature matching in object recognition.

3.2.3 Performance Evaluation

The performance of the proposed two-step feature detection algorithm and the feature matching with self-scaling mechanism is demonstrated in experiments in this section. The camera Bumblebee®2 shown in Fig. 3.13(a) is applied. The experimental setup is shown in Fig. 3.13(b). The camera running at a frame rate of 15 fps@640 × 480 pixels is fixed during the experiment, while the object with different locations is tested. For performance comparison, the standard SURF-based feature detection and feature matching are implemented in the experiment.

The snapshots of the experimental results are shown in Fig. 3.14. The distance of the object to the camera varies from 1.0 m, 3.0 m to 4.5 m, respectively. The matching results are summarized in Tab. 3.2 and in Fig. 3.15. Compared with the results based on SURF, the proposed approach shows a increased matching rate even in long distance. For example, with the conventional SURF-based approach, the matching rate at the distance of 4.5 m is only 1.5% as shown in the bottom left frame in Fig. 3.14, while with the proposed approach the matching rate is significantly increased to 23.8%, see the bottom right frame in Fig. 3.14. More importantly, the error of the feature matching is successfully avoided with the proposed approach. As shown in the bottom left image in Fig. 3.14, incorrect matching pairs appear (denoted by dashed blue-lines) with the standard approach. However, with the proposed approach the matching error is eliminated, see the bottom right image in Fig. 3.14.

Based on the experimental results it is concluded that the proposed image processing algorithm can increase the matching rate and reduce the matching error, and thus improve the robustness of the feature matching results. Compared with the conventional approach, it enables additionally object detection when the object is far away from the camera (e.g. ≥ 4 m). Thus, the proposed algorithm opens the possibility of object tracking at a wide spatial range.

3.3 The Latency of the System

Since the latency in the feedback is critical to the performance and the stability of the system, the system latency of NVSC system is analyzed and modeled in this section.

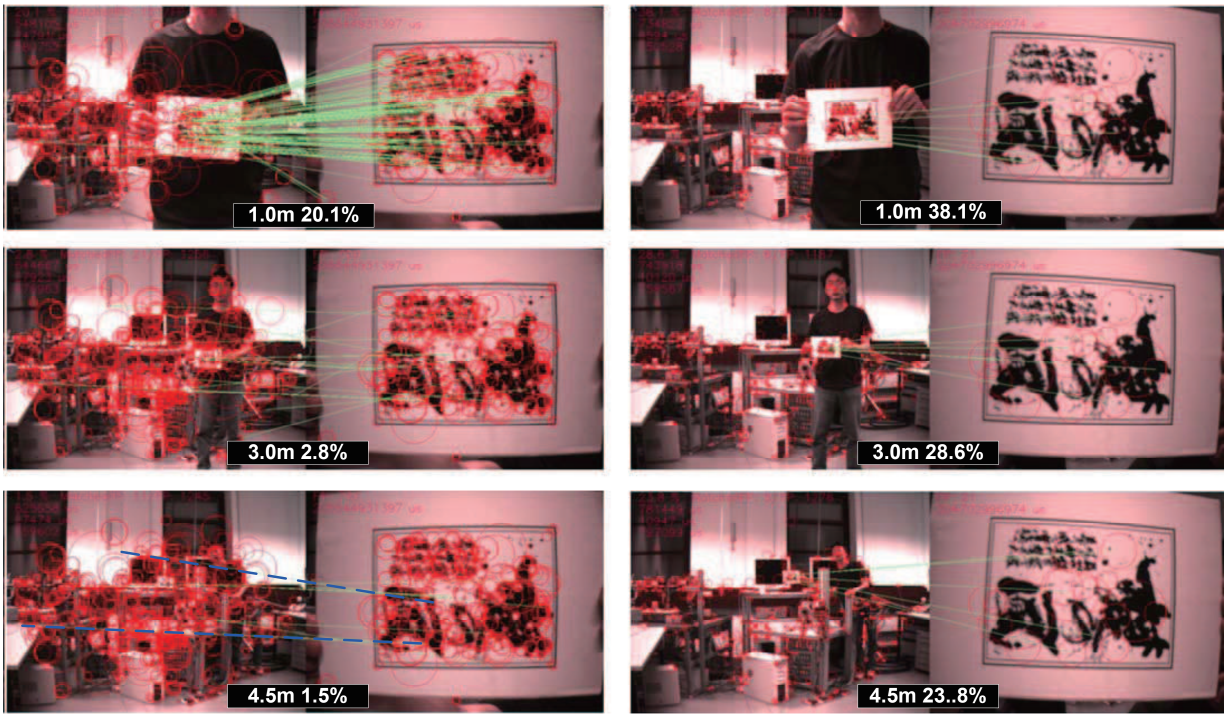


Figure 3.14: Performance comparison of feature matching based on SURF (left column) and proposed approach with binomial distributed sampling in scale space and self-scaling mechanism (right column). For each image, left side is the current image frame, and right side is the reference image fame. Object distance: 1.0 m, 3.0 m, and 4.5 m. The matching rate for the left column is (top down): 20.1%, 28%, and 1.5%, while for the right column is: 38.1%, 28.6%, and 23.8%. The dashed blue-lines in the bottom left image denote the incorrect matching pairs.

Distance [m]	Matching rate	
	proposed approach	standard approach
1.0	38.1%	20.1%
3.0	28.6%	2.8%
4.5	23.8%	1.5%

Table 3.2: Matching results of the proposed approach (with self-scaling mechanism) and the standard approach.

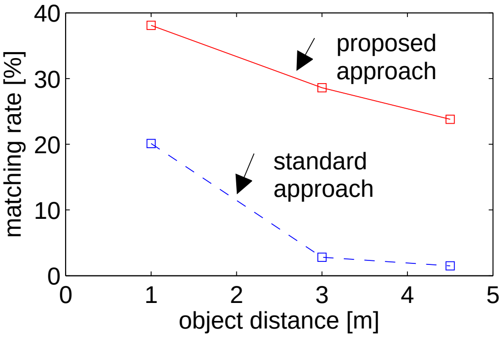


Figure 3.15: Comparison of the matching rate.

In this thesis, a NVSC system with tracking objective is studied. The block diagram of the system is shown in Fig 3.16. The sensor data, e.g. images, are captured and sent over network with sending interval d_k to the processing node, where d_k denotes the sending interval at time instant t_k. After receiving the raw sensor data, data processing is carried out to extract useful information. The processing results are sent to controller, which determines the control signal for actuators. The image processing results are also

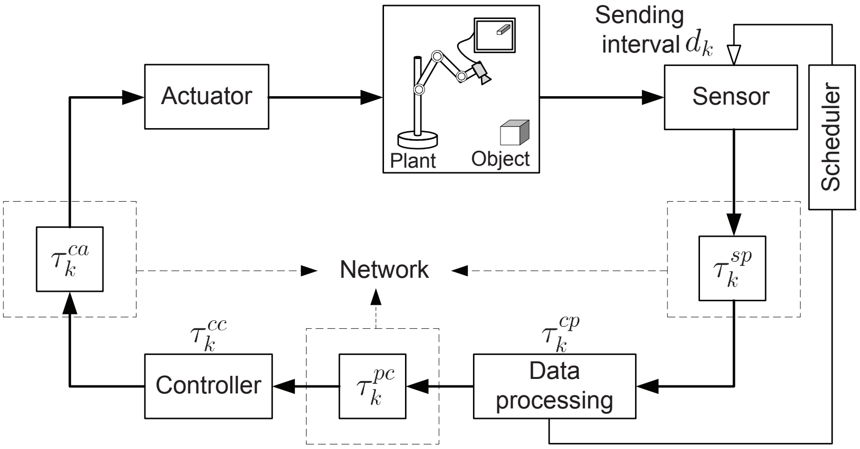


Figure 3.16: System diagram of NVSC system with varying feedback delays. d_k: sending interval at time instant t_k; τ_k^{sp}: transmission delay from sensor to processing node; τ_k^{cp}: computation delay on processing node; τ_k^{pc}: transmission delay from processing node to controller; τ_k^{c}: computation delay on controller; τ_k^{ca}: transmission delay from controller to actuator.

utilized by a scheduler which adjusts the sending interval d_k online for saving the network bandwidth. The details about the design of the scheduler will be presented in Chapter 4.

The overall feedback delay of systems shown in Fig. 3.16 consists of several parts:

- τ_k^{sp}: transmission delay from sensor to processing node,
- τ_k^{pc}: transmission delay from processing node to controller,
- τ_k^{ca}: transmission delay from controller to actuators,
- τ_k^{cp}: computation delay for data processing on processing node,
- τ_k^{cc}: computation delay for calculating control signals on controller.

For NVSC systems considered in this work, the transmission delay τ_k^{sp} is mainly caused by transmitting image data from the sensor node to the processing nodes. Since there are large volume image data needed to be transmitted, in general the image transmission delay τ_k^{sp} is significantly larger than the time delays τ_k^{pc} and τ_k^{ca} caused by transmitting image processing results and control signals, which are only several bytes. Besides, the computational cost for image processing largely exceeds the computation time for computing control signals. Therefore, during system design it is focused on the image transmission time delay and the computation time delay caused by image processing. The other three types of delays are omitted in the following. For simplicity of notation, τ_k^{x} and τ_k^{c} are adopted to replace τ_k^{sp} and τ_k^{cp}. The resulted system is shown in Fig. 3.17.

For system modeling and control design, the properties of the image processing delay τ_k^{c}, the data transmission time delay τ_k^{x}, and the sending interval d_k are discussed below.

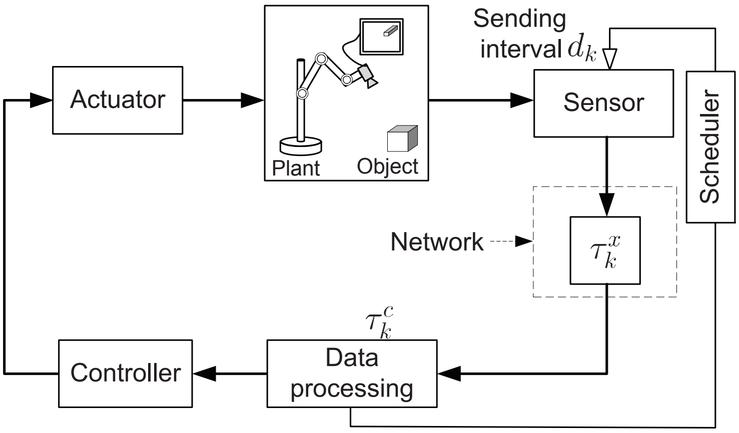


Figure 3.17: System diagram of NVSC system with varying sending interval d_k, transmission delay τ_k^x and computation delay τ_k^c.

3.3.1 Computation and Transmission Time delays

The image processing time delay is primarily caused by feature extraction, feature matching, and pose estimation. As each image contains a different scene and thereby different features, the feature number varies from frame to frame due to different view angle, illumination conditions, and random noise. In this thesis, the varying feature number over time is modeled as a random independent and identically distributed (i.i.d.) sequence resulting in a random i.i.d. image processing time delay τ_k^c. Since the i.i.d. process is a popular model in the communication community [182], an i.i.d. transmission time delay τ_k^x is considered in this thesis. Furthermore, the communication network / computational load is assumed such that the transmission/computation time delay is lower and upper bounded:

$$\tau_k^x \in [\underline{\tau}^x, \bar{\tau}^x], \quad \tau_k^c \in [\underline{\tau}^c, \bar{\tau}^c]. \tag{3.2}$$

For further analysis, it is assumed that the computation delays and the transmission delays take values in finite sets:

$$\begin{aligned} \tau_k^c &\in \mathrm{T}^c = \{\mathrm{T}^{c1}, \mathrm{T}^{c2}, \dots, \mathrm{T}^{cp}\}, \quad p \in \mathbb{N}, \\ \tau_k^x &\in \mathrm{T}^x = \{\mathrm{T}^{x1}, \mathrm{T}^{x2}, \dots, \mathrm{T}^{xq}\}, \quad q \in \mathbb{N}. \end{aligned} \tag{3.3}$$

3.3.2 Sending Interval

The sending interval between the t_k-th and the t_{k+1}-th sampling is denoted by d_k. For the tracking problem studied in this work, the tracking error is utilized for sending rate scheduling, which will be introduced in Chapter 4. Assume the tracking error is random i.i.d.. Accordingly, the sending interval d_k is modeled as i.i.d. variable. In oder to simplify the analysis it is assumed the sending interval takes values in a finite set

$$d_k \in D = \{d^1, \ d^2, \ \dots, \ d^M\}, \quad M \in \mathbb{N}, \tag{3.4}$$

where $d^m > d^{m+1}$, $m = 1, \dots, M-1$. In order to determine the values of $d^1, \dots, d^M$, it has to consider about the available computational resources and the computational complexity of the image processing. For example, with only one processing node, in order to guarantee that the image processing is finished before receiving the next image, the worst case processing time $\bar{\tau}^c$ is selected as the smallest sending interval. With more processing nodes available, a smaller sending interval can be assigned, e.g. with two processing nodes a sending interval $\frac{\bar{\tau}^c}{2}$ can be assigned. Assume there are a maximum of G processing nodes available for image processing. The lower bound of the sending interval becomes

$$\underline{d} = d^M = \frac{\bar{\tau}^c}{G}.$$

The upper bound of the sending interval relies on a heuristic choice. It is chosen as the upper bound of the computation time delay in this work

$$\bar{d} = d^1 = \bar{\tau}^c.$$

Therefore, the sending interval d_k is bounded by $d_k \in [\underline{d}, \bar{d}]$, where

$$\underline{d} = \frac{\bar{\tau}^c}{G}, \quad \bar{d} = \bar{\tau}^c. \tag{3.5}$$

The values of $d^2, \ldots, d^{M-1}$ can be arbitrarily selected. In this work, we associate them with the number of processing nodes in the following way

$$\{d^2 = \frac{\bar{\tau}^c}{2}, d^3 = \frac{\bar{\tau}^c}{3}, \ldots, d^m = \frac{\bar{\tau}^c}{g}, \ldots, d^{M-1} = \frac{\bar{\tau}^c}{G-1}\}, \tag{3.6}$$

where g denotes the number of necessary processing nodes for the sending interval d^m. The policy of selecting the appropriate sending interval d_k from (3.4) and the associated number of processing nodes g will be discussed in Section 4.3.

In this section, the property of the feedback delays within the cloud computing platform has been discussed. The efficacy of this platform equipped with ncRTP, i.e. how well the control performance is improved based on the parallel image processing, will be presented in the following chapter after the introduction of the control design and the stability analysis.

3.4 Discussion

For highly dynamic vision-based motion control high sampling rate visual feedback is demanded. With the advances in communication and information technology, utilizing networked computational resources for data processing in particular for image processing becomes an attractive solution as an alternative to traditional data processing on specific hardware or supercomputer. By distributing images to processing nodes connected over the communication network, parallel image processing (cloud computing) is realized, which thus accelerates visual feedback for vision-based control system. Apart from the sampling rate of visual feedback, the accuracy of visual perception is also essential for high-performance visual servo control. The visual perception provides information about the surroundings and the objects, and thus needs to be reliable and robust to different pose and illumination variance.

In this chapter networked high-speed vision system and robust image processing algorithm, including feature detection and feature matching, are systematically investigated. In order to establish the networked vision system, a real-time transport protocol ncRTP for large volume image data transmission is developed. The main issues of transport protocol design, e.g. fragmentation, congestion control and synchronization, are addressed. Based on ncRTP a cloud computing platform, consisting of a streaming server, several processing nodes, and a controller over the communication network, is established. With this platform high-sampling-rate visual feedback is achievable.

Robust image processing algorithm critical to system performance is concerned. A potential problem of standard feature matching is that the matching rate declines rapidly with increasing object distance, and matching error may appear. Methods are proposed to optimize the feature matching algorithm in the scale space level. A coarse evenly-spaced sampling in the whole scale space and a binomial sampling around the average object scale are cooperatively utilized for feature detection. The self-scaling mechanism, which suggests matching only around the scale of the object determined online, is adopted in the process of feature matching. Thus, the problem of low matching rate and matching error with large object distance is solved.

The important characteristic of the NVSC system, namely the system latency, is analyzed for the stability analysis and the control design to be introduced in Chapter 4. The i.i.d. process is applied to model the random transmission delay, image processing delay, and the sending interval.

The benefits of the proposed approaches introduced in this chapter are summarized below:

- The sampling rate of the visual feedback is increased through parallel image processing with computation resources connected over a communication network.
- The performance of feature matching is improved benefiting from optimized sampling distribution in scale space.

As the proposed networked vision system is capable of integrating the computation power of distributed processing nodes, and providing high-speed visual feedback, it acts as an ideal based for the development of high-speed visual servo control concerning the computation and the visual perception capabilities. Yet, the processing platform with multi-sensor systems and image processing methods with more advanced filter operator are not addressed and will be subject to feature research.

4 Stochastic NVSC with Random Delay

In the preceding chapter a distributed computation platform has been established for parallel image processing. A real-time transport protocol ncRTP has been developed to realize the transmission of large volume image data within NVSC systems. Methods have been proposed to improve the performance of image processing considering the sampling distribution in scale space. This chapter is concerned with stability analysis, control design, cost-performance trade-off for NVSC systems.

As discussed in Chapter 3, a NVSC system has long time delay caused by image processing in the feedback loop. The system suffers from additional transmission delay resulting from distributed image processing over a communication network. The feedback delays degrade the control performance, and may even render the system unstable. The closed loop behavior is further impacted by random dropouts of packages during data transmission. Besides, a NVSC system, which requires a non-equidistant sampling for saving the communication bandwidth, has aperiodic sampling intervals. Control approaches have been proposed to deal with NCSs considering the network-induced delays. In this thesis, the focus is on networked visual control system with random computation and communication time delays as well as aperiodic sampling intervals.

High-sampling-rate visual feedback, which guarantees good control performance, is obtained with many processing nodes run in parallel. Yet, the resources of a communication network such as bandwidth are limited. Therefore, in the presence of limited resources, keeping the data transmission over the communication at a minimum under the constraint of sufficient control, e.g. a *cost-performance trade-off*, is required. In order to utilize the network resources more efficiently, it is necessary to design a smart sending rate scheduling which adjusts the image sending interval online; meanwhile, the control performance is ensured. In this work, a network and system co-design approach is proposed to cope with the cost-performance trade-off.

The primary goal of this chapter is the stochastic analysis of NVSC system concerning the stability, the performance, and the cost. Contributions are a novel control law switching upon the overall feedback delay, stability analysis with the consideration of stochastic feedback delays and apeirodic sampling, and sending rate scheduling mechanisms for saving network resources. Key challenges are system modeling and formulation regarding the time-varying feedback time delays, the stability analysis, and the formulation of the cost function concerning the performance and the cost.

The reminder of this chapter is organized as follows: Firstly, the system modeling of NVSC systems with varying delays and the reformulation of the systems intro continuous-time system are introduced in Section 4.1. The stability analysis and switching controller design algorithms are presented for NVSC systems in Section 4.2. The trade-off between control performance and network load due to image data transmission is addressed in Section 4.3. Finally, the chapter is closed with summary and discussion in Section 4.4.

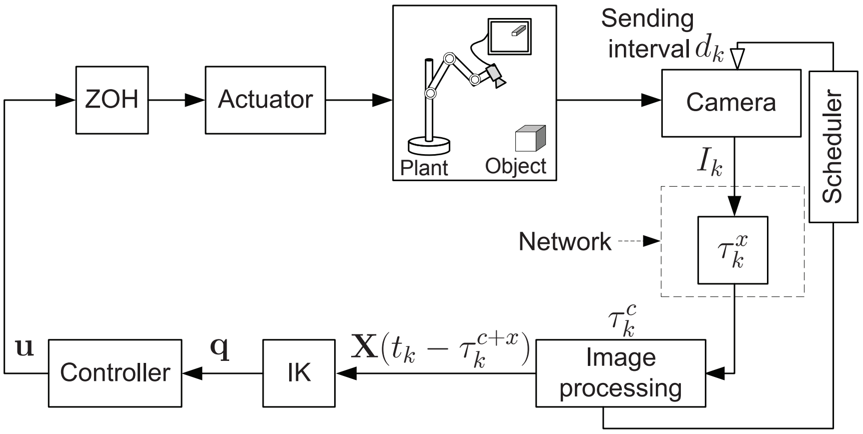


Figure 4.1: System diagram of networked position-based visual servo control for object tracking. I_k: image frame captured by the camera at time instant t_k; $\mathbf{X} \in \mathbb{R}^6$: pose of the end-effector relative to the reference object in Cartesian space; τ_k^{c+x}: feedback time delay consisting of image processing time delay τ_k^c and data transmission time delay τ_k^x; $\mathbf{q}$: joint displacement; $\mathbf{u}$: control signal; IK: inverse kinematics; d_k: sending interval; t_k: sampling instant; ZOH: zero-order hold.

4.1 Problem Definition

A 3D tracking problem with NVSC based on the cloud computing platform developed in Chapter 3 is studied in this work. Position-based visual servoing (PBVS) is applied and a camera-in-hand structure is selected for object detection and localization. The design issues about system modeling and system reformulation for NVSC systems are discussed in this section.

4.1.1 System Modeling

The system diagram of NVSC with position-based approach is shown in Fig. 4.1. In order to approximate an event-triggered vision system, a camera running at a high image frame rate, e.g. $\geq 400\,\text{Hz}$, is selected in this work. Compared to the time delays of data transmission and image processing typically of $\geq 30\,\text{ms}$, the image sampling interval, here $\leq 2.5\,\text{ms}$, is negligible. Besides, the image transmission over the communication network is triggered by a sending rate scheduler. Thus, the vision system is approximately considered as an event-triggered sampling system.

The image sequence captured by the eye-in-hand camera is denoted by I, and the image captured at sampling time t_k is denoted as I_k. The captured images are sent to the processing nodes over the communication network with the sending interval d_k. The NVSC system becomes a sampled-data system with zero-order hold (ZOH). After running the image processing on the cloud computing platform, the pose of the end-effector $\mathbf{X}(t_k - \tau_k^{c+x}) \in \mathbb{R}^6$ relative to the object is available to the controller with the time delay

$$\tau_k^{c+x} = \tau_k^c + \tau_k^x,$$

where τ_k^{c+x} denotes the feedback time delay for the t_k-th image consisting of the image processing time delay τ_k^c and the data transmission time delay τ_k^x. Through the robot inverse kinematic model, the Cartesian relative pose $\mathbf{X}$ is mapped on a joint displacement $\mathbf{q}(t_k - \tau_k^{c+x})$ in the joint space. Then, the control signal $\mathbf{u}$ is calculated in the joint space, and applied to the individual motors in the joints driving the robot motion. The feedback $\mathbf{X}(t_k - \tau_k^{c+x})$ is also utilized to adjust the sending interval d_k with the scheduler, as shown in Fig. 4.1. The policy of sending interval scheduling is presented in Section 4.3.

For the preliminary study of NVSC systems with time delays, a locally linearized version of the robot dynamics is considered by the computed torque feed-forward approach [183] (see also Appendix A.2) resulting in the linear time-invariant continuous-time plant

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + B\mathbf{u}(t),$$

with initial condition $\mathbf{x_0}$, where $\mathbf{x}(t) = [\mathbf{q}\ \dot{\mathbf{q}}]^T \in \mathbb{R}^n$ is the state vector and $\mathbf{u}(t) \in \mathbb{R}^n$ is the control signal; $\mathbf{q}$ denotes the joint angles; A, B are constant matrices of appropriate dimensions. Assuming sampling of the states at the instant t_k the state feedback controller is given by

$$\mathbf{u}(t) = K\mathbf{x}(t_k), \quad t \in [t_k, t_{k+1}), \quad \forall k \in \mathbb{N},$$

and the closed loop system is derived as

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + BK\mathbf{x}(t_k), \quad t \in [t_k, t_{k+1}). \tag{4.1}$$

For feedback systems with periodic sampling and ideal data transmission channels, the closed loop system (4.1) is equidistantly updated by

$$h = t_{k+1} - t_k.$$

In this case, the lifting technique [122] is applied to derive stability conditions and appropriate control algorithms. However, for NVSC systems the feedback time delays depend on data transmission time delay, the image processing time delay, and the sending interval, which are random and time-varying.

As introduced in Section 3.3, the value of the state vector $\mathbf{x}(t_k)$ arrives at the controller side with a feedback time delay τ_k^{c+x} consisting of the image processing and data transmission time delays, as shown in Fig. 4.2. At the controller side the holding interval between two consecutive updates at time instants $t_k + \tau_k^{c+x}$ and $t_{k+1} + \tau_{k+1}^{c+x}$ is denoted by h_k

$$h_k = t_{k+1} + \tau_{k+1}^{c+x} - t_k - \tau_k^{c+x} = d_k + \tau_{k+1}^{c+x} - \tau_k^{c+x}. \tag{4.2}$$

Throughout this work, it is assumed that packets do not overtake each other, meaning that packets arrive at the controller according to their sending order.

As a result, the closed loop system (4.1) becomes a sampled-data system with the feedback delay τ_k^{c+x}

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + BK\mathbf{x}(t_k), \quad t \in [t_k + \tau_k^{c+x},\ t_{k+1} + \tau_{k+1}^{c+x}). \tag{4.3}$$

The control problem to be addressed in this work is formulated as follows:

Control problem: Given a system with random feedback time delays, develop a control algorithm such that the closed loop system in (4.3) is exponentially mean-square stable (EMSS), i.e. satisfying

$$\mathbb{E}\{||\mathbf{x}(t)||^2 | \mathbf{x_0}\} \leq b||\mathbf{x_0}||^2 e^{-\rho(t-t_0)},$$

where $b > 0$, $\rho > 0$ are real numbers and $\mathbf{x_0}$ is the initial condition.

For stability analysis and control design, the discrete-time system in (4.3) is further reformulated into a continuous-time system.

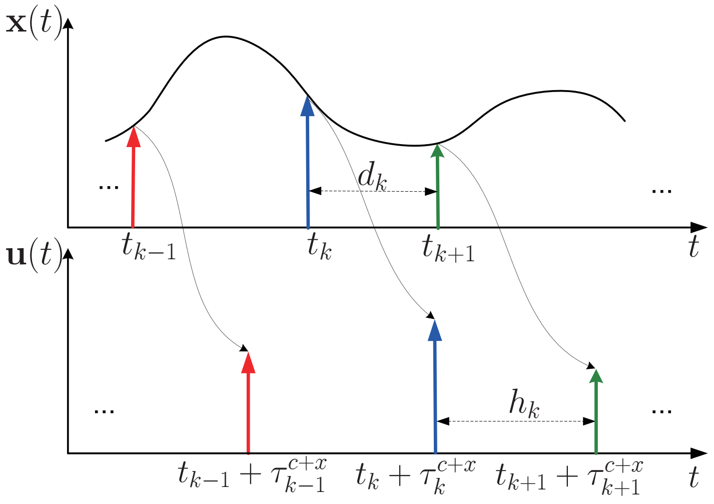


Figure 4.2: Time delay diagram of NVSC systems. τ_k^c: computation time delay for image I_k sampled at time instant t_k; τ_k^x: transmission time delay for image I_k from sensor to processing node; $\tau_k^{c+x} = \tau_k^c + \tau_k^x$; d_k: sending interval between the t_k−th and t_{k+1}−th samplings; h_k: holding interval between two consecutive updates at the controller side.

4.1.2 System Reformulation for Control Design

In this section, the NVSC system with random sending interval, random computation and transmission time delays is reformulated into a time-varying delay system by means of the input-delay approach [184]. In order to achieve stability and high control performance, a switching control mechanism is proposed which switches depending on the value of the overall time delay.

Reconsider the sending instant t_k as

$$t_k = t - (t - t_k) = t - \tau(t), \quad t \in [t_k + \tau_k^{c+x},\ t_{k+1} + \tau_{k+1}^{c+x}),$$

where $\tau(t)$ is the overall time delay for the processed sensory feedback $\mathbf{x}(t_k)$ during the period $[t_k + \tau_k^{c+x}, t_{k+1} + \tau_{k+1}^{c+x})$ at the controller side, see Fig. 4.3. The overall feedback time delay $\tau(t)$ consists of three parts: the computation time delay τ_k^c, the transmission time delay τ_k^x, and the holding time delay $h(t)$

$$\tau(t) = \tau_k^c + \tau_k^x + h(t) = \tau_k^{c+x} + h(t), \quad t \in [t_k + \tau_k^{c+x},\ t_{k+1} + \tau_{k+1}^{c+x}), \tag{4.4}$$

where

$$h(t) \in [0,\ h_k), \quad t \in [t_k + \tau_k^{c+x},\ t_{k+1} + \tau_{k+1}^{c+x}),$$

with h_k given by (4.2). Between two consecutive updates at the controller side the holding delay $h(t)$ has the slope

$$\dot{h}(t) = 1,$$

as shown in Fig. 4.3. Combining (3.2), (3.5), (4.2) and (4.4), the overall time-varying delay is bounded by $\underline{\tau} \leq \tau < \bar{\tau}$ with

$$\underline{\tau} = \underline{\tau}^c + \underline{\tau}^x, \quad \bar{\tau} = \bar{d} + \bar{\tau}^c + \bar{\tau}^x = 2\bar{\tau}^c + \bar{\tau}^x.$$

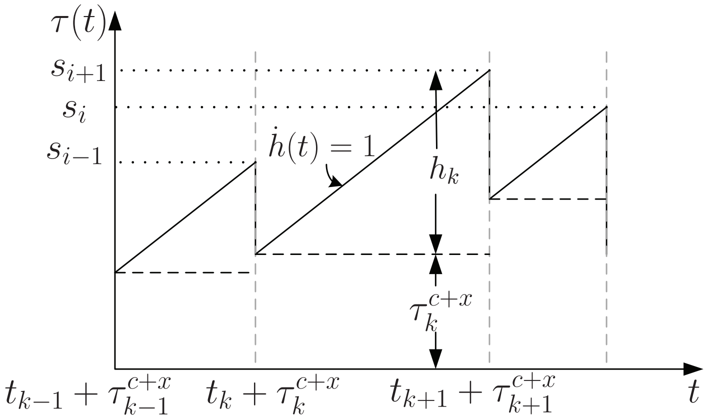


Figure 4.3: The evolution of time-varying time delay $\tau(t)$.

Substituting $\mathbf{x}(t_k) = \mathbf{x}(t - \tau(t))$ into the closed loop system in (4.3) yields

$$\begin{aligned} \dot{\mathbf{x}}(t) &= A\mathbf{x}(t) + BK\mathbf{x}(t-\tau(t)), \quad t \in [t_k + \tau_k^{c+x},\ t_{k+1} + \tau_{k+1}^{c+x}), \\ \mathbf{x_0} &= \mathbf{x}(\theta), \quad \theta \in [-\bar{\tau}, 0], \end{aligned} \tag{4.5}$$

where $\mathbf{x}(\theta)$ is the initial condition. Hence, the sampled-data system with time-varying sampling intervals is reformulated into a continuous-time system with time-varying feedback time delay $\tau(t)$.

Remark 2 If packet dropout appears during data transmission, the previous received data is hold and utilized by the system (4.3) until a new packet is received. In this case, the impact of packet dropout can be viewed as accumulated transmission delay between two updates at the controller side. Assume the maximum number of consecutive packet dropouts is a, the the transmission delay τ^x is bounded by

$$\tau_k^x \in [\underline{\tau}^x,\ (a+1)\bar{\tau}^x],$$

see also Fig. 4.4

4.2 Switching Control Approach for NVSC

In order to stabilize the closed loop system and at the same time provide high control performance, a time delay-dependent switching control approach is proposed, in which the state feedback control gain switches according to the current time delay value. The time-varying time delay is categorized by boundary values s_i into n finite intervals

$$\begin{aligned} S_1 &= \{\tau | s_0 \leq \tau < s_1\}, \\ S_2 &= \{\tau | s_1 \leq \tau < s_2\}, \\ &\vdots \\ S_n &= \{\tau | s_{n-1} \leq \tau < s_n\}, \end{aligned} \tag{4.6}$$

where $0 < s_i < s_{i+1}$, for $i = 1, \ldots, n-1$, and $s_0 = \underline{\tau}$ and $s_n = \bar{\tau}$, see also Fig. 4.3. The delay distribution is discretized to reduce the computational complexity for the control design, as for each discrete value of time delay a different control gain needs to be computed.

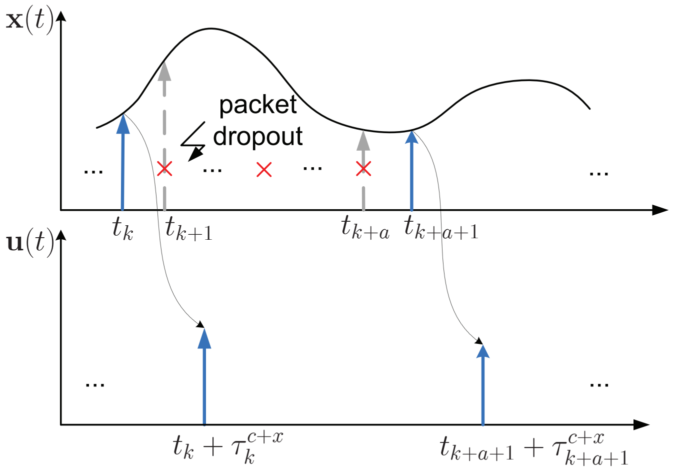


Figure 4.4: Delay diagram of NVSC systems with packets dropout during data transmission. The maximum number of consecutive packet dropout is denoted by a.

The switching control law is defined by

$$\mathbf{u}(t) = K_{i(\tau(t))}\mathbf{x}(t - \tau(t)), \quad i = 1, \dots, n, \tag{4.7}$$

where $K_{i(\tau(t))}$ denotes the state feedback controllers switching depending on the value of the time delay $\tau(t)$. Accordingly, the system in (4.5) can be rewritten as

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + \sum_{i=1}^{n} \beta_i B K_i \mathbf{x}(t - \tau(t)), \tag{4.8}$$

where β_i is a delay-dependent indicator function defined by

$$\beta_i = \begin{cases} 1, & s_{i-1} \leq \tau < s_i, \quad i = 1, \dots, n \\ 0, & \text{otherwise,} \end{cases} . \tag{4.9}$$

and the dependence of the index i of $(\tau(t))$ is omitted for simplicity of notation. The resulting system (4.8) is a continuous-time system with a controller switching the feedback gains depending on the time-varying feedback time delay.

Compared with the conventional non-switching control approach which applies a constant gain designed for the highest appearing time delay $\bar{\tau}$, the proposed switching control is less conservative with respect to control performance [185]. The delay categorization, the stability analysis, and the control design are given in the following.

4.2.1 Delay Categorization

The statistical properties of category bounds s_i are important for stability analysis. Before carrying out the stability analysis, the selection of appropriate category bounds is discussed in this section.

Remember that random sending interval d_k, transmission delay τ_k^x and computation delay τ_k^c are modeled by i.i.d. process in Section 3.3. For the stability analysis and control design the discrete time delay values then act as upper bounds for their corresponding discretization. According to the convolution law of independent random variables [186], select s_i in (4.6) as any subset of S

$$s_i = \tau_k^{c+x} + h_k = \tau_{k+1}^{c+x} + d_k, \quad s_i \in S \subset \{\mathrm{T}^c + \mathrm{T}^x + D\},$$

with T^c, T^x and D defined in (3.3) and (3.4). Then s_i is also i.i.d., see also Fig. 4.3. It implies that the indicator function β_i in (4.9) is a binary i.i.d. process, i.e. has a Bernoulli distribution. As a result,

$$\Pr\{\beta_i = 1\} = p_i, \quad \sum_{i=1}^{n} p_i = 1. \tag{4.10}$$

Its expected value and variation are given by

$$\mathbb{E}\{\beta_i\} = p_i, \quad \mathbb{E}\{(\beta_i - p_i)^2\} = p_i(1 - p_i).$$

In order to derive the occurrence probability of p_i from the occurrence probabilities of the delay components, the time-varying time delays τ^x, τ^c and d_k are categorized into $U \geq 1$, $V \geq 1$ and $W \geq 1$ intervals with s_u^c, s_v^x and s_w^d satisfying

$$\begin{aligned} s_{u-1}^c \leq \tau_k^c < s_u^c, \quad & s_u^c > 0, \quad u = 1, \ldots, U, \\ s_{v-1}^x \leq \tau_k^x < s_v^x, \quad & s_v^x > 0, \quad v = 1, \ldots, V, \\ s_{w-1}^d \leq d_k < s_w^d, \quad & s_w^d > 0, \quad w = 1, \ldots, W, \end{aligned} \tag{4.11}$$

with s_u^c, s_v^x and s_w^h taking values in the sets T^c, T^x and D respectively. Further assume that the occurrence probabilities of the delay intervals are

$$\begin{aligned} \Pr\{s_{u-1}^c \leq \tau_k^c < s_u^c\} = p_u, \quad & \sum_{u=1}^{U} p_u^c = 1, \\ \Pr\{s_{v-1}^x \leq \tau_k^x < s_v^x\} = p_v, \quad & \sum_{v=1}^{V} p_v^x = 1, \\ \Pr\{s_{w-1}^d \leq d_k < s_w^d\} = p_w, \quad & \sum_{w=1}^{W} p_w^d = 1. \end{aligned} \tag{4.12}$$

The delay intervals in (4.6) and the associated occurrence probabilities of indicator function in (4.10) becomes

$$s_i = \sum_{u=1}^{U} \sum_{v=1}^{V} \sum_{w=1}^{W} s_u^c + s_v^x + s_w^d, \quad p_i = \sum_{u=1}^{U} \sum_{v=1}^{V} \sum_{w=1}^{W} p_u^c \, p_v^x \, p_w^d. \tag{4.13}$$

Remark 3 Note that a higher resolution of the discretization results in lower conservatism of the control design regarding the performance. The performance improves as the number of delay intervals s_i and also the number of state feedback controllers K_i is increased. However, a large number of delay intervals will result in higher computational complexity. In particular, the dimension of LMIs increases proportionally with the number of delay intervals as shown later.

The switching control mechanism discussed above aims at improving the control performance of NVSC systems with varying feedback delays. Based on the analysis of delay intervals s_i and the associated occurrence probability p_i in this section, the stability analysis and a controller design approach are introduced below.

4.2.2 Stability Analysis

The objective of this section is to derive a mean exponential stability condition for the system in (4.8). A Lyapunov-Krasovskii approach is selected to analyze the stability of system (4.8) as it is stochastic and constrains time delays. In order to derive a delay-dependent condition, the following Newton-Leibnitz formula is considered.

$$\int_{t-s_i}^{t} \dot{x}(s)ds = x(t) - x(t-s_i). \tag{4.14}$$

Substituting the Newton-Leibnitz formula into (4.8) and defining $z^T(t) = [x^T(t)\ \dot{x}^T(t)]$ results in the closed-loop system

$$E\dot{z}(t) = \bar{A}z(t) - \sum_{i=1}^{n} \bar{A}_i \int_{t-s_i}^{t} z(s)ds, \tag{4.15}$$

where

$$E = \begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}, \quad \bar{A} = \begin{bmatrix} 0 & I \\ A + \sum_{i=1}^{n} \beta_i B K_i & -I \end{bmatrix}, \quad \bar{A}_i = \begin{bmatrix} 0 & 0 \\ 0 & \beta_i B K_i \end{bmatrix}. \tag{4.16}$$

The system (4.15) is used for analysis and controller synthesis. The stability condition is represented by an easy computable LMI conditions as given in detail in the following theorem.

Theorem 1 The closed-loop system (4.8) is mean exponentially stable, if there exist symmetric matrices, $Q_i > 0$, $i = 1, \ldots, n$, $P_1 > 0$ and real matrices P_2 and P_3 with

$$P = \begin{bmatrix} P_1 & 0 \\ P_2 & P_3 \end{bmatrix}, \tag{4.17}$$

such that the following LMI is satisfied

$$\begin{bmatrix} \Psi & s_1 P^T & \ldots & s_n P^T \\ * & -s_1 Q_1 & 0 & \vdots \\ \vdots & 0 & \ddots & * \\ * & \ldots & * & -s_n Q_n \end{bmatrix} < 0, \tag{4.18}$$

where

$$\begin{aligned}\Psi &= \begin{bmatrix}\Xi_1 & \Xi_2\\ P_1 - P_2 & -P_3\end{bmatrix} + \begin{bmatrix}\Xi_1 & \Xi_2\\ P_1 - P_2 & -P_3\end{bmatrix}^T\\ &\quad + \sum_{i_1}^{n} s_i \begin{bmatrix}0 & 0\\ 0 & p_iBK_i\end{bmatrix}^T Q_i \begin{bmatrix}0 & 0\\ 0 & p_iBK_i\end{bmatrix},\\ \Xi_1 &= A^TP_2 + \sum_{i=1}^{n} p_i(BK_i)^TP_2,\\ \Xi_2 &= A^TP_3 + \sum_{i=1}^{n} p_i(BK_i)^TP_3.\end{aligned} \tag{4.19}$$

The LMI stability condition in Theorem 1 can be efficiently solved by computational toolbox for Matlab, e.g. `Yalmip` [187].

Proof: Before the proof is shown, the following definition and lemma have to be given.

Definition 1 [188] Let $\mathcal{L}$ be the infinitesimal generator of a function $V(z(t))$. Then, the operator $\mathcal{L}$ acting on $V(z(t))$ is defined as

$$\mathcal{L}V(z(t)) = \lim_{\Delta\to 0}\frac{1}{\Delta}\left\{\mathbb{E}\{V(z(t+\Delta))|z(t)\} - V(z(t)\right\}.$$

Lemma 1 [189] Let X and Y be real constraint matrices with appropriate dimensions. Then

$$X^TY + Y^TX \le \varepsilon X^TX + \frac{1}{\varepsilon}Y^TY$$

holds for any $\varepsilon > 0$.

Lemma 2 [190] Suppose M_{11}, M_{12}, M_{21}, M_{22} are $p\times p$, $p\times q$, $q\times p$ and $q\times q$ matrices respectively, M_{22} is invertible. Let

$$M = \begin{bmatrix}M_{11} & M_{12}\\ M_{21} & M_{22}\end{bmatrix},$$

such that M is a $(p+q)\times(p+q)$ matrix. Then the *Schur Complement* of the block M_{22} of the matrix M is the $p\times p$ matrix

$$S = M_{11} - M_{12}M_{22}^{-1}M_{21}.$$

Lemma 3 [191] If β is nonnegative and u satisfies the integral inequality

$$u(t) \le \alpha(t) + \int_0^t \gamma(s)u(s)ds,$$

and if function α is constant, then

$$u(t) \le \alpha e^{\int_0^t \gamma(s)ds},$$

resulting in *Gruonwall-Bellman* lemma.

Consider a Lyapunov-Krasovskii functional candidate

$$V(z(t)) = V_0(z(t)) + \sum_{i=1}^{n} V_i(z(t)),$$

where

$$\begin{aligned} V_0(z(t)) &= z^T(t)EPz(t), \\ V_i(z(t)) &= \int_{-s_i}^{0}\int_{t+\Theta}^{t} z^T(s)\bar{A}_i^T Q_i \bar{A}_i z(s) ds d\Theta. \end{aligned}$$

According to Definition 1, it has

$$\begin{aligned} \mathcal{L}V_0(z(t)) =& \dot{z}^T(t)EPz(t) + z^T(t)P^T E\dot{z}(t) \\ =& z(t)\bar{A}Pz(t) - \sum_{i=1}^{n}\int_{t-s_i}^{t} z^T(s)ds\bar{A}_i^T Pz(t) \\ &+ z^T(t)P^T\bar{A}z(t) - \sum_{i=1}^{n} z^T(t)P^T\bar{A}_i\int_{t-s_i}^{t} z(s)ds \\ =& z^T(t)[\bar{A}^T P + P^T\bar{A}]z(t) \\ &- \sum_{i=1}^{n}\int_{t-s_i}^{t}[z^T(t)P^T\bar{A}_i z(s) + z^T(s)\bar{A}_i^T Pz(t)]ds. \end{aligned}$$

According to Lemma 1, $\mathcal{L}V_0(z(t))$ becomes

$$\begin{aligned} \mathcal{L}V_0(z(t)) \leq& z^T(t)[\bar{A}^T P + P^T\bar{A}]z(t) \\ &+ \sum_{i=1}^{n}\int_{t-s_i}^{t}[z^T(t)P^T Q_i^{-1}Pz(t) + z^T(s)\bar{A}_i^T Q_i\bar{A}_i]ds \\ \leq& z^T(t)[\bar{A}^T P + P^T\bar{A}]z(t) + \sum_{i=1}^{n} s_i z^T(t)P^T Q_i^{-1}Pz(t) \\ &+ \sum_{i=1}^{n}\int_{t-s_i}^{t} z^T(s)\bar{A}_i^T Q_i\bar{A}_i z(s)ds. \end{aligned} \tag{4.20}$$

Likewise, it has

$$\sum_{i=1}^{n}\mathcal{L}V_i(z(t)) = \sum_{i=1}^{n} s_i z^T(t)\bar{A}_i^T Q_i\bar{A}_i z(t) - \sum_{i=1}^{n}\int_{t-s_i}^{t} z^T(s)\bar{A}_i^T Q_i\bar{A}_i z(s)ds. \tag{4.21}$$

Combine (4.20) and (4.21), it yields

$$\begin{aligned} \mathcal{L}V(z(t)) \leq& z^T(t)[\bar{A}^T P + P^T\bar{A} + \sum_{i=1}^{n} s_i\bar{A}_i^T Q_i\bar{A}_i \\ &+ \sum_{i=1}^{n} s_i P^T Q_i^{-1}P]z(t) \\ =& z^T(t)\Theta z(t). \end{aligned} \tag{4.22}$$

Apply Schur complement given in Lemma 2 to (4.22), it results in (4.18).

Note that $\max_{\theta\in[-\tau,0]}\{||z(t+\theta)||\} \leq \phi||z(t)||$ for some $\varphi > 0$, the following inequality can be established

$$\begin{aligned} V(z(t)) \leq& \left[\lambda_{\max}(EP) + \phi\sum_{i=1}^{n}\frac{s_i^2}{2}\lambda_{\max}(\bar{A}_i^T Q_i \bar{A}_i)\right]||z(t)||^2 \\ \leq& \Lambda_{\max}||z(t)||^2. \end{aligned} \tag{4.23}$$

Combining (4.22) and (4.23) yields

$$\frac{\mathcal{L}V(z(t))}{V(z(t))} \leq -\frac{\lambda_{\min}(-\Theta)}{\Lambda_{\max}} \triangleq -\rho_0, \tag{4.24}$$

and

$$\mathbb{E}\{\mathcal{L}V(z(t))\} \leq -\rho_0\mathbb{E}\{V(z(t))\}. \tag{4.25}$$

By applying Dynkin's formula into (4.25) it becomes

$$\begin{aligned} \mathbb{E}\{V(z(t))\} - \mathbb{E}\{V(z(0))\} =& \mathbb{E}\left\{\int_0^t \mathcal{L}V(z(s))ds\right\} \\ \leq& -\rho_0\int_0^t \mathbb{E}\{V(z(s))\}ds. \end{aligned} \tag{4.26}$$

Using the Gronwall-Bellman lemma defined in Lamma 3, (4.26) results in

$$\mathbb{E}\{V(z(t))\} \leq e^{-\rho_0 t}\mathbb{E}\{V(z(0))\}. \tag{4.27}$$

Since

$$\begin{aligned} V(z(t)) \geq& \left[\lambda_{\min}(EP) + \sum_{i=1}^{n}\frac{s_i^2}{2}\lambda_{\min}(Q_i)\right]||z(t)||^2 \\ =& \Lambda_{\min}||z(t)||^2, \end{aligned} \tag{4.28}$$

it is established that

$$\mathbb{E}\{||z(t)||^2\} \leq e^{-\rho_0 t}\frac{\mathtt{E}\{V(z(0))\}}{\Lambda_{\min}}. \tag{4.29}$$

Eq. (4.29) provides the proof for mean exponential stability. ■

Hence, the system in (4.8) is demonstrated to be mean exponential stable.

4.2.3 Controller Design

Solving feedback gains K_i, $i = 1, ..., n$ in Theorem 1 involves nonlinear terms, e.g. $P_2^T BK_1$, $P_3^T BK_1$, $P_2^T BK_2$ and $P_3^T BK_2$ in (4.19). These nonlinear terms render the inequality in (4.18) into a bilinear matrix inequality (BMI) problem, whose solutions are difficult to find as it is non-convex and NP-hard. However, the nonlinear terms can be eliminated by choosing a special matrix $X = P^{-1}$ and so an LMI formulation is recovered. The controller design algorithm is given in the following theorem.

Theorem 2 Given positive scalars r_1 and r_2, if there exist symmetric matrices $R_i > 0$, $i = 1, ..., n$ and $X_1 > 0$ satisfying

$$X = \begin{bmatrix} X_1 & 0 \\ -r_1 X_1 & r_2 X_1 \end{bmatrix},$$

such that

$$\begin{bmatrix} \hat{\Psi} & \hat{\Psi}_1^T & \cdots & \hat{\Psi}_n^T \\ * & -s_1 R_1 & 0 & \vdots \\ \vdots & 0 & \ddots & * \\ * & \cdots & * & -s_n R_n \end{bmatrix} < 0, \tag{4.30}$$

where

$$\hat{\Psi} = \begin{bmatrix} -r_1 X_1 & r_2 X_1 \\ \Xi_3 & -r_2 X_1 \end{bmatrix} + \begin{bmatrix} -r_1 X_1 & r_2 X_1 \\ \Xi_3 & -r_2 X_1 \end{bmatrix}^T + \sum_{i=1}^{n} s_i R_i,$$

$$\Xi_3 = A X_1 + \sum_{i=1}^{n} p_i B Y_i + r_1 X_1,$$

$$\hat{\Psi}_1 = s_1 \bar{A}_1 X = s_1 \begin{bmatrix} 0 & 0 \\ -p_1 r_1 B Y_1 & p_1 r_2 B Y_1 \end{bmatrix},$$

$$\vdots$$

$$\hat{\Psi}_n = s_n \bar{A}_n X = s_n \begin{bmatrix} 0 & 0 \\ -p_n r_1 B Y_n & p_n r_2 B Y_n \end{bmatrix},$$

holds, then the NVSC system is MES with the feedback gain

$$K_i = Y_i X_1^{-1}, \quad i = 1, \ldots, n. \tag{4.31}$$

Proof: Define

$$X = P^{-1} = \begin{bmatrix} X_1 & 0 \\ -r_1 X_1 & r_2 X_1 \end{bmatrix}.$$

Pre- and post-multiply Θ in (4.22) by X^T and X, it becomes

$$\bar{A} X + X^T \bar{A}^T + \sum_{i=1}^{n} s_i Q_i^{-1} + \sum_{i=1}^{n} s_i X^T \bar{A}_i^T Q_i \bar{A}_i X < 0. \tag{4.32}$$

Let $R_i = Q_i^{-1}$ and $Y_i = K_i X_1$, $i = 1, \ldots, n$. Applying Schur complement to (4.32) results in (4.30). ■ ■

Remark 4 The structure of matrix X is chosen based on the requirement $P^{-1} = X$, where $EP = P^T E$. Therefore, X is determined as follows

$$X = \begin{bmatrix} X_1 & 0 \\ X_2 & X_3 \end{bmatrix}, \quad X_1 = X_1^T > 0. \tag{4.33}$$

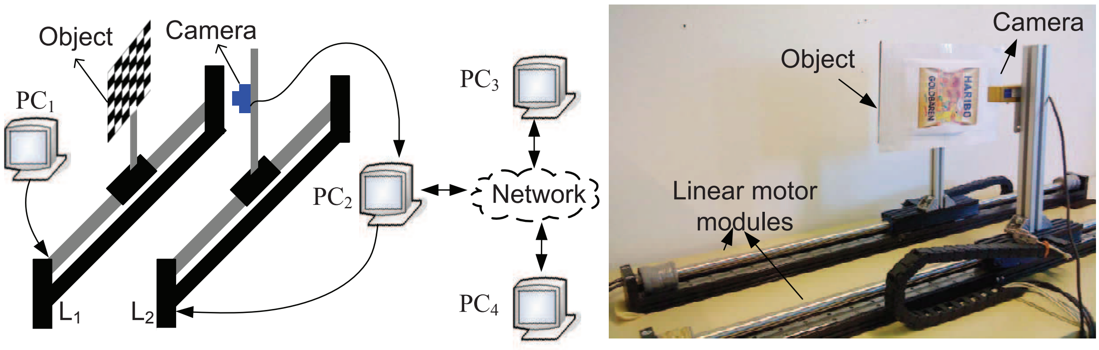


Figure 4.5: Experimental setup of a NVSC system with distributed computation. L_1, L_2: linear motor modules; PC$_1$: PC for control of the linear motor module carrying the object; PC$_2$: PC for image capturing, image streaming, and control of the linear module carrying the camera; PC$_3$ and PC$_4$: PCs for image processing, connected to PC$_2$ over a communication network.

However, by expanding $\hat{\Psi}_i$, $i = 1, ..., n$ in (4.30) it results in terms, e.g. BK_iX_j, $i = 1, ..., n$ and $j = 1, 2, 3$, which make deriving an LMI formulation impossible. In order to obtain an LMI formulation, X_2 and X_3 in (4.33) are restricted to $-r_1X_1$ and r_2X_1, where r_1 and r_2 are positive real numbers. Although LMI can be efficiently solved by LMI toolbox, the restriction on matrix X introduces certain conservatism in the controller design and leads to unfeasibility in some cases. When the unfeasibility occurs, the general form of X in (4.33) is used and an BMI (bilinear matrix inequality) solver is applied to solve the problem in (4.30). Some numerical examples are given in Appendix B.1.

Performance Evaluation

In order to validate the efficacy of the distributed computation platform developed in Chapter 3 and the proposed switching control law introduce above, experiments are carried out on two commercial 1-DOF linear motor modules from Copley Controls, as shown in Fig. 4.5. The control objective is to drive the linear motor module which carries a camera to track the moving object on the reference linear motor module.

Experimental Setup Both linear motor modules are controlled through MATLAB/SIMULINK blocksets on two standalone PCs (PC$_1$ for controlled module and PC$_2$ for reference module) with a sampling rate of 1KHz. The reference module controlled by PC$_1$ (i686, AMD, Athlon(tm), Processor 3000+) is assigned with the following signal:

$$x_d(t) = 0.15\,\sin(1.57\,t).$$

The camera, Mikrotron EoSens MC1363 (see parameters of the camera in Appendix A.1), is connected to PC$_2$ (X86-64, AMD, Phenom IIx4 810 Processor) with the controller implemented on the same PC. The camera runs at a frame rate of 60 fps with a resolution of 640 × 480 pixels. Other two PCs, PC$_3$ (i686, AMD, Phenom(tm), 9850, Quard-Core Processor) and PC$_4$ (i686, AMD, Phenom(tm) IIx4 945 Processor) equipped with NVIDIA GeForce 8800 GTX graphic cards, are selected for image processing. The image data transmission from PC$_2$ to PC$_3$/PC$_4$ is realized based on ncRTP developed in Chapter 3.

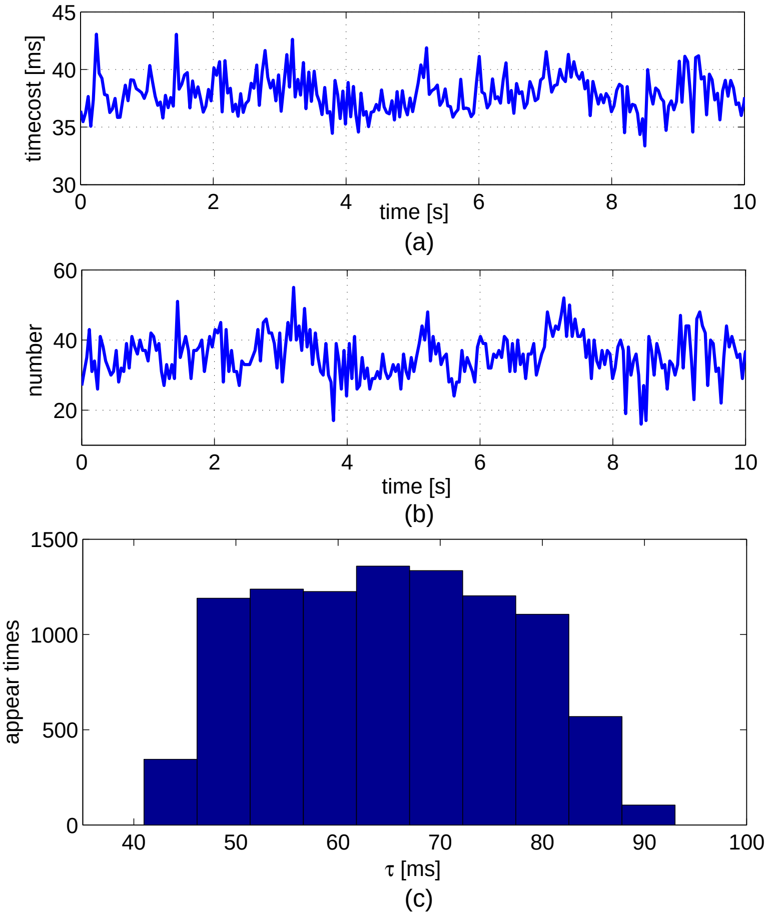


Figure 4.6: Image processing time (a), number of extracted image features (b) and histogram of delays in the feedback loop (c) in the experiment.

An image captured by the camera at time instant t_k is packetized with a timestamp when the image is buffered and sent from PC_2 to either PC_3 or PC_4. The image processing results together with the timestamp are sent from both PCs to the controller on PC_2 through the communication network. The timestamp is then extracted from the received data. By comparing it with the current time the overall feedback delay including data transmission, image processing and holding delays is obtained.

Feedback Delay The time required for real-time image processing depends on image features and has random values between 33ms to 45ms. The relationship between image features and image processing delay is shown in Fig. 4.6 (a) (b). The cross correlation coefficient between these two signals is 0.93, which indicates that image processing delay is closely related with image feature number. More image features require more time for image processing, e.g. 18 image features take about 33ms while 55 image features take about 43.5ms. As soon as the image processing is finished, a new image is acquired and processed. This results in a random sampling interval up to 45ms. In addition, random delay of data transmission is simulated by a network emulator (Netem, see also Appendix B.2) having i.i.d. delay ranging from 5ms to 10ms. The overall delay (i.e. the

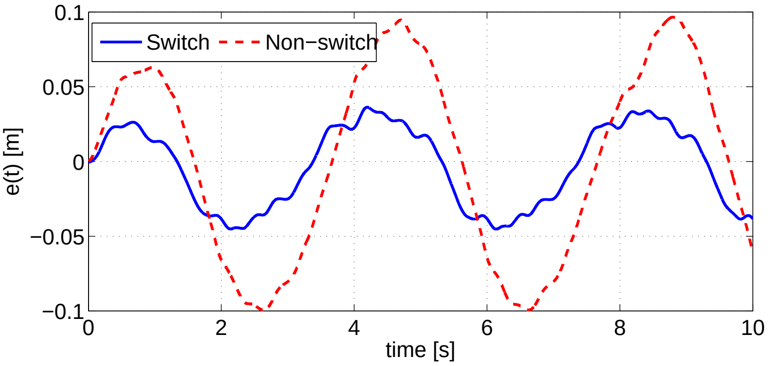


Figure 4.7: Mean control error evolution of delay-dependent controller (solid line) and standard design controller (dash line).

sum of computation delay, transmission delay and sampling interval) has the range from 38 ms to 100 ms as shown in the Fig. 4.6 (c).

Control Design Applying the controller design approach in Section 4.2.3 and considering two delay intervals $n = 2$ with $s_1 = 65\,\text{ms}$, $p = 0.53$ selected heuristically, the feedback gains K are

$$K_1 = 800, \quad K_2 = 160.$$

The experiment runs 20 times with the same initial conditions of both modules. For the comparison of control performance, the standard non-switching controller is implemented, i.e. a robust controller that stabilizes for all occurring time delays up to $\bar{\tau} = 100\text{ms}$. The corresponding controller design gain is

$$K = 160.$$

Experimental Results The control error is defined by

$$e(t) = x_d(t) - x_c(t), \tag{4.34}$$

where $x_d(t)$ and $x_c(t)$ denote the position measurements of the reference module and the controlled module respectively. The resulting control errors are shown in Fig. 4.7, and also summarized in Tab. 4.4.

Table 4.1: Performance comparison.

	delay interval	$\|\|e\|\|_{\max}$ [cm]	$\|\|e\|\|_{\text{var}}$ [cm^2]
switch	2	3.63 (-62.4%)	7.22
non-switch	1	9.66	42.95

The delay dependent switching controller approach has maximal absolute tracking error $||e||_{\max} = 3.63\,\text{cm}$ and the variance of the absolute tracking error $||e||_{\text{var}} = 7.22\,\text{cm}^2$, while the maximal tracking error of standard design controller approach is $||e||_{\max} = 9.66\,\text{cm}$ and the variance of the tracking error $||e||_{\text{var}} = 42.95\,\text{cm}^2$. The experimental results demonstrate that the switching controller approach results in a better control performance compared to the conventional non-switching control design.

In this section, the continuous-time system in (4.8) is demonstrated to be stable with the desired delay-dependent switching controller. The proposed approach is validated in experiments. The results demonstrate performance benefits of the proposed design approach over the conventional counterpart. Another issue of NVSC systems, namely the network load due to the data transmission, has so far not been addressed yet. In the following section, the optimization problem of the trade-off between the control performance and network load will be discussed.

4.3 Cost-Performance Trade-Off for NVSC

In this section, the communication network load problem resulting from large volume image data transmission in NVSC systems is discussed. The goal of this section is to find the trade-off between the control performance and the communication network load of the NVSC system. Scheduling mechanisms are proposed to reduce the communication network load, and meanwhile to guarantee the control performance. Within the context of this work, sending rate scheduling means to i) determine an appropriate sending interval d_k at time instant t_k, and ii) to achieve desired optimal occurrence probabilities of different sending intervals. The details of the designed sending rate scheduling algorithm meeting these two requirements are introduced in this section.

4.3.1 Cost Function

Remember in (4.12) the sending interval is categorized into W intervals, and each interval has an occurrence probability p_w^d, $w = 1, \ldots, W$, which is closely related with the network usage. In order to determine the optimal occurrence probabilities of p_w^d, $w = 1, \ldots, W$, we consider here a cost function which jointly optimizes the communication network load and the control performance

$$J = \lim_{T\to\infty} \mathbb{E}\left\{ \int_0^T \mathbf{x}^T(t) R \mathbf{x}(t) dt + \frac{1}{T} \int_0^T C(d_k(t)) dt \right\}, \tag{4.35}$$

where $R \in \mathbb{R}^{n\times n}$, $R > 0$, T is the run period, and $C(d_k(t))$ denotes the communication network cost associated with the sending interval

$$C(d_k(t)) = C_w, \quad \text{if } s_{w-1}^d \leq d_k < s_w^d, \quad w = 1, \ldots, W,$$

where $C_w \in \mathbb{R}$ is the communication network cost factor associated with the sending interval d_k falling in the category $s_{w-1}^d \leq d_k < s_w^d$. In order to optimize the trade-off between the control performance and the network cost, the cost function in (4.35) is analyzed in the following.

Control Cost

The first term on the right-hand side in (4.35) represents the control performance. Given the boundary values s_i of the delay intervals, and assume the probability distributions of transmission and computation delays are constant and known, the expected value of the control performance is shown to be bounded by $\bar{J}_{\text{per}}(p_1^d, \ldots, p_W^d)$.

Proof: Consider the same Lyapunov-Krasovskii candidate in Section 4.2.2. Due to the fact $x(t) = [I\ 0]z(t)$, consider Dynkin's formula and eq. (4.22), define the performance cost function as

$$J_{per} = \mathbb{E}\bigg\{ \int_0^T x^T(t)Rx(t)dt \bigg\},$$

it becomes

$$\begin{aligned} J_{per} =& \mathbb{E}\bigg\{ \int_0^T z^T(t) \begin{bmatrix} I \\ 0 \end{bmatrix} R \begin{bmatrix} I\ 0 \end{bmatrix} z(t)dt \bigg\} \\ =& \mathbb{E}\bigg\{ \int_0^T \Big[z^T(t) \begin{bmatrix} I \\ 0 \end{bmatrix} R \begin{bmatrix} I\ 0 \end{bmatrix} z(t) + \mathcal{L}V(z(t))\Big]dt \bigg\} \\ &- \mathbb{E}\bigg\{ \int_0^T \mathcal{L}V(z(t))dt \bigg\} \\ \leq& \mathbb{E}\bigg\{ \int_0^T z^T(t)\bar{\Theta}z(t)dt + V(z(0)) \bigg\}, \end{aligned}$$

where $\bar{\Theta} = \Theta + \begin{bmatrix} I\ 0 \end{bmatrix}^T R \begin{bmatrix} I\ 0 \end{bmatrix}$. By the requirement $\bar{\Theta} < 0$, it is clear that

$$J_{per} = \mathbb{E}\bigg\{ \int_0^T x^T(t)Rx(t)dt \bigg\} \leq V(z(0)) = \bar{J}_{per}(p_1, \ldots, p_n),$$

and

$$\begin{aligned} \lim_{T\to\infty} \mathbb{E}\bigg\{ \int_0^T x^T(t)Rx(t)dt \bigg\} &\leq \bar{J}_{per}(p_1, \ldots, p_n) \\ &= V_0(z(0)) + \sum_{i=1}^{n} p_i V_i(z(0)). \end{aligned} \tag{4.36}$$

■

Note that the guaranteed cost in (4.36) is a function of $p_1, \ldots, p_n$. Assume the probability distributions of transmission and computation delays, $p_1^c, \ldots, p_U^c$ and $p_1^x, \ldots, p_V^x$ defined in (4.12), are constant and known. According to (4.13), the guaranteed cost (4.36) is determined by the probability distributions of sending intervals. The formula in (4.36) can be rewritten as

$$\lim_{T\to\infty} \mathbb{E}\bigg\{ \overset{T}{\underset{0}{\inf}}\, x^T(t)Rx(t)dt \bigg\} \leq \bar{J}_{\text{per}}(p_1^d, \ldots, p_W^d). \tag{4.37}$$

Thus, the control cost is demonstrated to be upper bounded by $\bar{J}_{\text{per}}$ conditioned on the occurrence probability distribution of sending intervals.

Network Cost

The second term denotes the communication network cost over the run time T. Assume that the communication network cost is a monotonically decreasing function of the sending interval. With a smaller sending interval $d_k \in [s_{w-1}^d,\ s_w^d)$, a higher communication network cost factor C_w is assigned. With the occurrence probability of each sending interval defined in (4.12), the expected value of the normalized communication network usage becomes

$$\lim_{T\to\infty} \mathbb{E}\bigg\{ \frac{1}{T} \int_0^T C(d_k(t))dt \bigg\} = \sum_{w=1}^{W} p_w^d C_w. \tag{4.38}$$

By combining equations (4.37) and (4.38), the stochastic cost function in (4.35) is bounded from above by a deterministic function

$$J(p_1^d, \dots, p_W^d) \leq \bar{J}_{\text{per}}(p_1^d, \dots, p_W^d) + \sum_{w=1}^{W} p_w^d C_w, \tag{4.39}$$

depending on the probability distribution of the sampling intervals. Minimizing (4.39) results in an optimal trade-off between the control performance and the communication network cost.

Observe that any change in p_w^d results in variations of p_i, see (4.13). And the stability condition in Theorem 1 for the system in (4.8) depends also on the probability distributions of the sending interval p_i, $i = 1, \dots, n$. Therefore, stability constraints have to be considered in the optimization problem. It becomes a static optimization problem with linear matrix inequality constraints. The details of determining the optimal occurrence probabilities of sending intervals are formulated as follows.

Proposition 1 An optimal probabilistic sending is given by

$$\begin{aligned} &\min_{p_1^d,\dots,p_W^d} J(p_1^d, \dots, p_W^d), \\ &\text{s.t.(LMI in Theorem 1)} \end{aligned} \tag{4.40}$$

where $p_1^d, \dots, p_W^d$ satisfying $\sum_{w=1}^{W} p_w^d = 1$ is the set of admissible probability distribution of sending intervals.

Hence, by solving (4.40) the probability distributions of different sending intervals are determined. A local minimum can be found by the function `fmincon` from the optimization toolbox in Matlab. The sending-rate scheduling methodologies with known $p_1^d, \dots, p_W^d$ will be introduced in the following section.

4.3.2 Optimal Random Sending Rate Scheduling

The objective of sending rate scheduling is to determine the processing node for the image I_t captured by the camera at time instant t_k. As the optimal occurrence probabilities of different sending intervals can be determined by optimizing the cost function (4.40) in Proposition 1, a random sending rate scheduling policy is designed in this section which meets the optimization results.

Policy of Sending Rate Scheduling

Remember there are M sending intervals in the set D in (3.4) ready for sending rate scheduling. The algorithm of sending rate scheduling realized at the streaming server side is described in Algorithm 1 and in Fig. 4.8:

- Firstly, at time instant t_k a sampling interval d_k is randomly selected from the set D, satisfying the determined occurrence probabilities p_w^d, for $s_{w-1} \leq d_k < s_w$, $w = 1, \dots, W$.
- Then, the number g of necessary processing nodes is determined according to (3.5)(3.6).

```
Algorithm 1 Image distribution with optimal random sending rate scheduling
Require: $d_k$
Task: send image $\mathrm{I}_t$ (image sampled at time instant $t$, $t \in [t_k,\ t_k + d_k g)$) to processing
  node with determined $d_k$
  select $d_k$ randomly s.t. optimal occurrence probabilities of different sending rate→ determine $g$ according to (3.5)(3.6).
  for t = $[t_k,\, t_k + d_k g)$ do
    $t_k = t$;
    send $\mathrm{I}_{t_k}$ to PC, (PC $\in \{\mathrm{PC}_1, \dots, \mathrm{PC}_g\}$ represents the processing node available at $t_k$);
    $t = t_k + d_k$;
  end for
```

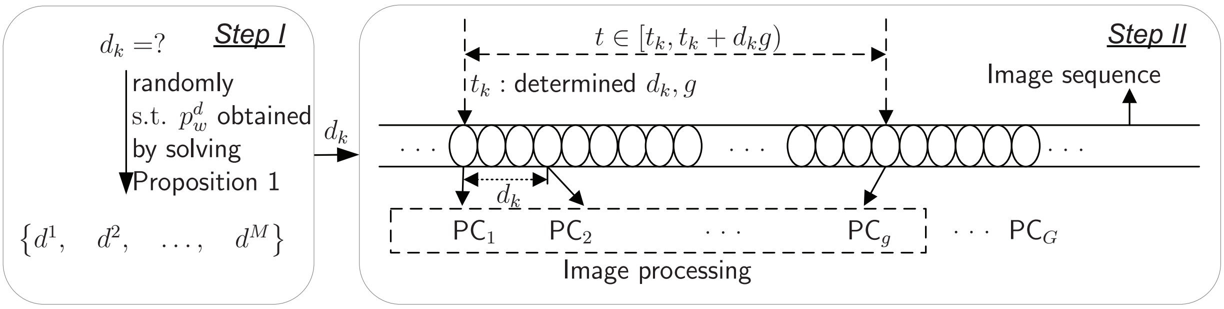


Figure 4.8: Sending rate scheduling at the streaming server side with optimal occurrence probabilities of the sending intervals. Left part: random selection of d_k considering optimal occurrence probability p_w^d, $w = 1, \dots, W$. Right part: image distribution with determined sending interval d_k, and the number of processing node is denoted by g.

- For the time period $t \in [t_k, t_k + d_k g]$, the image sequence with sampling interval d_k is sent to the processing nodes available over the communication network, also Fig. 4.8.
- After sending the images, the three steps above are repeated from the instant $t_k + d_k g$ and so on.

With the algorithm introduced above, an optimal sending rate scheduling is realized for NVSC systems with guaranteed control performance. Instead of transmitting every captured image, only selective images (sub-sampling of the original image sequence) are sent over the network with different sending intervals. Thus, the network load is reduced.

Performance Evaluation

In this section, the proposed NVSC with optimal sending rate scheduling is validated in experiments on two linear motor modules.

Experimental Setup The considered NVSC system consists of two linear motors, a camera and three PCs connected over network. The experimental setup is similar to that in Fig. 4.5. A target is mounted on a reference linear motor module assigned with a trajectory signal which has an amplitude of 20 cm and a frequency of $\pi/3$ rad/s. The two

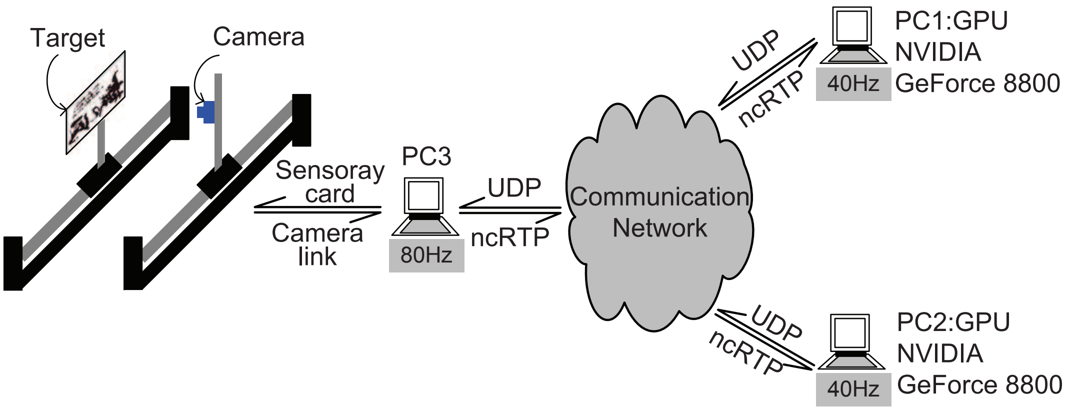


Figure 4.9: Experimental setup of a NVSC system comprising of a pair of 1-DoF linear motor modules (similar to experimental setup in Fig. 4.5). PC$_1$, PC$_2$: image processing with GPU implementation (40Hz). PC$_3$: image capturing (80Hz) and controller;

linear motor modules are connected to the host PC$_3$ running RTAI Linux via a Sensoray S626 I/O card, as shown in Fig. 4.9. In order to achieve high-speed visual feedback, the images captured by the camera with a resolution of 640 × 480 pixels are processed by distributed processing nodes PC$_1$, PC$_2$ (X86-64 AMD Phenom II ×4 810 processor) over a communication network. The image data is transmitted over the network based on ncRTP introduced in Chapter 3.

Control Design The system parameters are obtained through standard least square identification of the response to square pulse input

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.959 & -1169.9 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t). \tag{4.41}$$

For simplicity, consider a single interval for the combination of the transmission and the computation delays $\tau_k^c + \tau_k^x$ with the occurrence probability

$$\begin{aligned} P\{\tau_k^c + \tau_k^x \leq s_1^c + s_1^x\} &= p_1^c p_1^x = 100\%, \\ s_1^c + s_1^x = &\bar{\tau}^{c+x} = 35 \text{ ms}, \end{aligned}$$

where $\bar{\tau}^{c+x}$ is determined in the experiments. Select $s_1^d = \frac{1}{80}$ ms, $s_2^d = \frac{1}{40}$ ms, and according to (4.13), the delay intervals and the probability distributions become

$$\begin{aligned} s_1 &= s_1^c + s_1^x + s_1^d = 47.5 \,\text{ms}, & p_1 &= p_1^c p_1^x p_1^d = p_1^d, \\ s_2 &= s_1^c + s_1^x + s_2^d = 60 \,\text{ms}, & p_2 &= p_1^c p_1^x p_2^d = p_2^d. \end{aligned}$$

The probability distributions, p_1^d and p_2^d, are designed such that the optimal network utilization is achievable. The control module is equipped with a set of delay-dependent PD controllers. Combining the switching PD controller with (4.41) yields

$$\frac{d}{dt}\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.959 & -1169.9 \end{bmatrix} \begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} + \sum_{i=1}^{2} \beta_i K_i \begin{bmatrix} x(t-s_i) \\ \dot{x}(t-s_i) \end{bmatrix}, \tag{4.42}$$

where $K_i = \begin{bmatrix} 0 & 0 \\ -K_{Pi} & -K_{Di} \end{bmatrix}$.

Set the parameters $C_1 = 2$, $C_2 = 1$ and $R = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$. The optimization problem in Proposition 1 is numerically solved by the optimization `fmincon` as well as `Yalmip toolbox` in Matlab. With the initial condition $[x(0)\ \dot{x}(0)]^T = [0\ 0]$, $\theta \in [-s_2, 0]$, the cost function in (4.35) is optimized $[p_1^d\ p_2^d] = [50\%\ 50\%]$ for $J = 1.67$. The associated stabilizing state-feedback gains are

$$K_1 = \begin{bmatrix} 0 & 0 \\ -900 & -15 \end{bmatrix}, \quad K_2 = \begin{bmatrix} 0 & 0 \\ -600 & -5 \end{bmatrix}.$$

For comparing the control performance, a standard NVSC system without data scheduling (high network load), e.g. $[p_1^d\ p_2^d] = [100\%\ 0\%]$, is implemented

$$\begin{aligned} \frac{d}{dt}\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} &= \begin{bmatrix} 0 & 1 \\ -0.959 & -1169.9 \end{bmatrix}\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 & 0 \\ -900 & -15 \end{bmatrix}\begin{bmatrix} x(t-s_2) \\ \dot{x}(t-s_2) \end{bmatrix}. \end{aligned}$$

Sending Rate Scheduling In the experiments, the camera works at a frame rate of 80 Hz. As mentioned before, each image frame with a resolution of 640 × 480 pixels has a data size of 300KB. A sampling rate of 80 Hz means 192 Mb/s data flow over the network. In order to reduce the network load, Algorithm 1 for data scheduling introduced in the previous section is applied. Since only two processing nodes are available in the experiments, the data scheduling algorithm is simplified as following:

- PC$_1$: Images with frame number {1,3,...} are sent to PC$_1$ periodically, which leads to a sampling rate 40 Hz.

- PC$_2$: The subset of images with frame number {2,4,6,...} is sent to PC$_2$ randomly, which leads to a sampling rate $\leq$40 Hz. In the experiment, the images sent to PC$_2$ are randomly selected with desired probability distribution.

- PC$_3$: The controller works at bounded sampling rate [40 Hz,80 Hz].

Experimental Results The control error defined in (4.34) is utilized for performance evaluation. A comparable control performance is achieved by the proposed approach, as shown in Fig. 4.10. The proposed approach with data scheduling has a maximum tracking error $||e||_{\text{max}} = 2.29\,\text{cm}$ and a mean tracking error $||e||_{\text{mean}} = 1.22\,\text{cm}$, similar to the maximal and mean tracking error of high sampling rate design approach (+2.5%), see Tab. 4.2. However, the network usage (data flow) is 25% less than the one with high sampling rate design approach. It is noticed that the resulted tracking error is relative large under stability constraint. Besides, the PD controller without considering the signal dynamics leads also to large tracking error. And the control performance is further deteriorated by non-ideal friction compensation and the lack of delay compensation strategy.

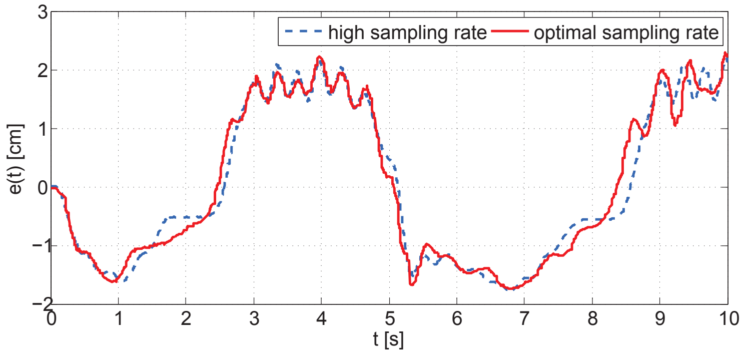


Figure 4.10: The mean control error evolution of the proposed approach with data scheduling $[p_1^d \; p_2^d] = [50\% \; 50\%]$ (solid line) and the standard approach without data scheduling $[p_1^d \; p_2^d] = [100\% \; 0\%]$ (dash line).

Table 4.2: Control performance and network usage.

sampling rate	$\|\|e\|\|_{\text{max}}$ [cm]	$\|\|e\|\|_{\text{mean}}$ [cm]	Network usage [unit]
optimal	2.29	1.22 (+2.5%)	1.5 (-25%)
high	2.19	1.19	2

In this section, a data scheduling algorithm has been designed for image streaming in NVSC systems. The proposed approach is validated by experiments on two linear motor modules. The results demonstrate comparable control performance of the proposed approach with less network cost compared with the conventional counterpart. So far only the optimal occurrence probabilities of sending intervals have been considered in the design of sending rate scheduling. The motion dynamics of the object that also restricts the tracking performance will be additionally integrated in sending rate scheduling in the following section.

4.3.3 Tracking Error based Sending Rate Scheduling

Instead of selecting the sending interval d_k randomly during the procedure, the tracking error is considered for sending rate scheduling. Generally speaking, a smaller sending interval should be assigned when the tracking error becomes larger. This is a reasonable heuristics for a visual servo control system with a camera-in-hand structure, because a large tracking error may result in losing the object in the image plane. However, by minimizing the cost function (4.35), the occurrence probability of each sending interval is predetermined, which places additional constraint on the scheduling. Therefore, for sending rate scheduling both the tracking error and the optimal sending rate probability are taken into consideration.

The proposed sending rate scheduling algorithm is shown in Fig. 4.11. Similar to the policy introduced in Section 4.3.2, there are two steps in sending rate scheduling: the first step is to determine the sending interval d_k at time instant t_k, while the second step is to distribute images to processing nodes with determined sending interval.

Policy of Sending Rate Scheduling

In order to reduce the communication network load due to the image data transmission, the tracking error $\mathbf{X}$ in Cartesian space is considered in selecting an appropriate sending interval at the steaming server side. Remember in Section 4.2.1 the sending interval d_k is categorized into W intervals, see eq. (4.11). In order to select the sending interval d_k according to the tracking error, the tracking error $\mathbf{X}$ is categorized also into W intervals

$$r_{w-1} \leq |\mathbf{X}| < r_w, \quad w = 1, \ldots, W, \tag{4.43}$$

where $r_w > r_{w-1} > 0$ are the bounds of the intervals, $r_0 = 0$ and $r_W \to \infty$.

As pointed out previously, both the tracking error X and the optimal occurrence probabilities p_w^d, $w = 1, \ldots, W$, are utilized for selecting the sending interval d_k at time instant t_k. Assume the tracking error falls into the interval $r_{w-1} \leq |\mathbf{X}| < r_w$. If till the time instant t_k the measured occurrence probability of the sending interval $s_{w-1} \leq d_k < s_w$, denoted by $p_w^d(t_k)$, is smaller than the pre-determined optimal occurrence probability p_w^d, the sending interval d_k is selected. Here, $d_k = d^m$, $d^m \in D$ is randomly selected from the set of the sending intervals falling in the interval $s_{w-1} \leq d^m < s_w$. Otherwise, another sending interval should be chosen from the sending intervals, whose optimal occurrence probabilities are not reached till the time instant t_k. Denote the set consisting of these sending intervals with $\underline{D}$, and it becomes a subset of D: $\underline{D} \subset D$. In order to select an appropriate sending interval, the set $\underline{D}$ is further divided into two groups

$$\begin{aligned} \underline{D}_1 &= \{d^{\underline{m}} | d^{\underline{m}} \in \underline{D}, d^{\underline{m}} < s_{w-1}\}, \\ \underline{D}_2 &= \{d^{\underline{m}} | d^{\underline{m}} \in \underline{D}, d^{\underline{m}} > s_w\}, \end{aligned}$$

where $d^{\underline{m}}$ represents the element in $\underline{D}$. Then, the sending intervals in $\underline{D}_1$ and $\underline{D}_2$ are assigned with different priorities:

$$\begin{aligned} &\mathrm{Pri}(\underline{D}_1) > \mathrm{Pri}(\underline{D}_2), \\ &\mathrm{Pri}(d^{\underline{m}}) < \mathrm{Pri}(d^{\underline{m}+1}), \quad d^{\underline{m}} \in \underline{D}_1, \\ &\mathrm{Pri}(d^{\underline{m}}) > \mathrm{Pri}(d^{\underline{m}+1}), \quad d^{\underline{m}} \in \underline{D}_2, \end{aligned}$$

where $\mathrm{Pri}(\cdot)$ denotes the priority of the set or the element during the selection of the sending interval, see also Fig. 4.11(a) for visualization. The rule of assigning the priority above is reasonable, since for a tracking problem studied in this work, the first task is to keep the object stay in the field of view.

Finally, the selection algorithm is summarized as

$$d_k = \begin{cases} d^m, & s_{w-1} < d^m < s_w, \quad \text{if} \quad p_w^d(t_k) < p_w^d, \\ d^{\underline{m}}, & d^{\underline{m}} \in \underline{D}, \quad \text{otherwise}, \end{cases} \tag{4.44}$$

where the sending interval $d^{\underline{m}}$ has the maximum priority of being selected among the elements in the set $\underline{D}$. After d_k is determined, the number of necessary processing nodes g can be found according to (3.5)(3.6).

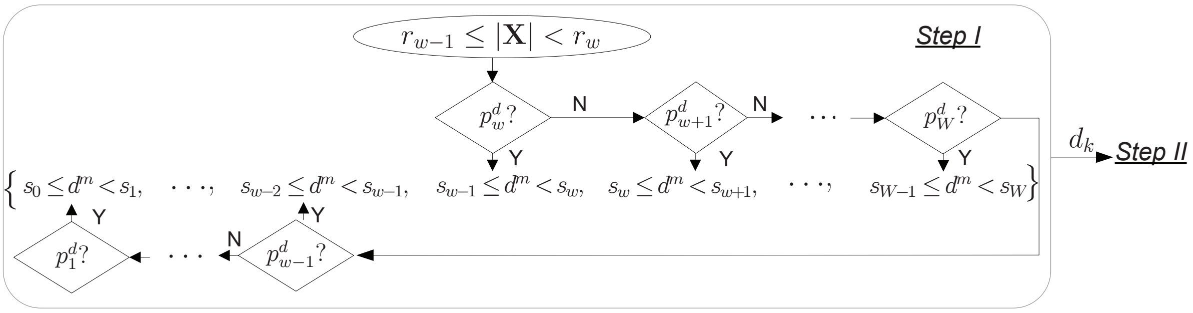

$r_{w-1} \leq |\mathbf{X}| < r_w$
Step I
p_w^d?
N
p_{w+1}^d?
N
$\cdots$
p_W^d?
Y
Y
Y
d_k
Step II
$\{s_0 \leq d^m < s_1, \quad \cdots, \quad s_{w-2} \leq d^m < s_{w-1}, \quad s_{w-1} \leq d^m < s_w, \quad s_w \leq d^m < s_{w+1}, \quad \cdots, \quad s_{W-1} \leq d^m < s_W\}$
Y
Y
N
p_1^d?
$\cdots$
p_{w-1}^d?

Figure 4.11: Sending rate scheduling based on the tracking error at the streaming server side. Step I: determination of the sending interval d_k considering the tracking error $\mathbf{X}$ and occurrence probability p_m^d at time instant t_k. The tracking error $\mathbf{X}$ is categorized into W intervals with r_w denoting the bounds of the intervals. Step II: given in Fig. 4.8.

Algorithm 2 Image distribution with tracking error based sending rate scheduling

```
Require: $d_k$
Task: send image $\mathrm{I}_t$ (image sampled at time instant $t$, $t \in [t_k,\ t_k + d_k g)$) to processing
  node with determined $d_k$
  sorting $|\mathbf{X}|$ with (4.43) $\rightarrow$ determine $d_k$ according to (4.44) $\rightarrow$ determine $g$ according
  to (3.5)(3.6).
  for t = $[t_k,\ t_k + d_k g)$ do
    $t_k = t$;
    send $\mathrm{I}_{t_k}$ to PC, (PC $\in \{\mathrm{PC}_1, \ldots, \mathrm{PC}_g\}$ represents the processing node available at $t_k$);
    $t = t_k + d_k$;
  end for
```

With determined sending interval d_k from Step I, the streaming server starts to send images to processing nodes. The details of image distribution are illustrated in Fig. 4.8 as well as in Algorithm 2, which is different from Algorithm 1 in terms of the method for selecting the sending interval d_k.

With the algorithm introduced above, the communication network load caused by image transmission is effectively reduced. Meanwhile, an acceptable control performance is maintained.

Remark 5 Note that in the current work, only the tracking error and the optimal occurrence probabilities of the sending interval are considered in the sending interval determination. In the future, network congestion and the packet loss rate should be integrated in design of sending rate scheduling as well.

Performance Evaluation

In order to validate the efficacy of the proposed sending rate scheduling algorithm, the proposed approach is compared with the standard non-scheduling approach. Experiments for a 3D tracking problem with a 14-DoF dual-arm [192] are conducted. The details of the robot arm are given in Appendix A.

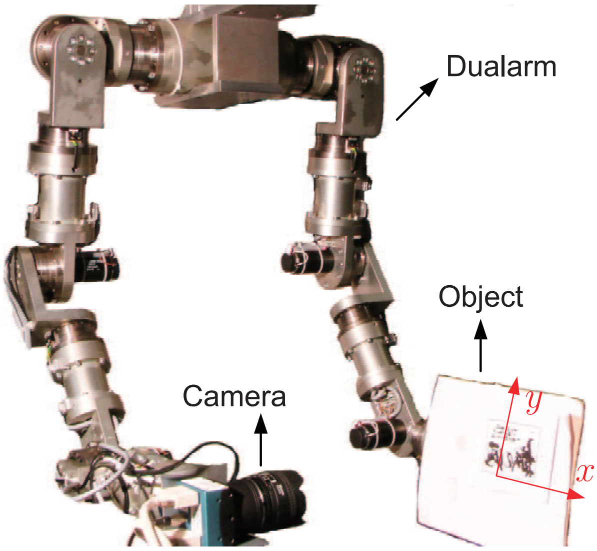


Figure 4.12: Experimental setup with 7-DoF robot arm, high-speed camera (mounted on the end-effector of the left arm) and moving object (mounted on the end-effector of the right arm).

Experimental Setup The end-effector of the right arm (7-DoF) is equipped with a high-speed camera (Mikrotron MC1363), while the object is mounted on the end-effector of the left arm (7-DoF), as shown in Fig.4.12. In the experiments, the camera runs at a high sampling rate of 400 Hz@640 × 480 pixels approximating an event-based sampling system. Two sending rates $\{40\,\mathrm{Hz}, 80\,\mathrm{Hz}\}$ are selected for sending rate scheduling. The dual-arm is connected to a PC running real-time RTAI/Linux. The control loop is implemented in MATLAB/SIMULINK blocksets. Standalone real-time code is generated directly from the SIMULINK models. Two other PCs, PC_1 and PC_2 run image processing in parallel.

Image Processing On the processing nodes, image processing algorithms for 3D pose estimation are implemented. First, the image features are extracted from the image. To increase the accuracy of the pose estimation and speed up visual feedback, a GPU implementation of SURF [168] is applied by exploiting its massive parallel processing capability. After feature matching [193], RANSAC (RANdom Sample Consensus) [194] is selected for outliers rejection. Fig. 4.13 shows an example of a matching result during the experiment. Finally, the relative pose between the camera and the object is obtained by calculating the extrinsic parameters (relative translation and rotation) of the camera based on the matched feature pairs.

Control Design After linearization of the manipulator equations through computed torque control the system can be represented by seven decoupled subsystems. For each joint, the overall time delay τ is categorized into two intervals with $s_1 = 45$ ms. By optimizing the cost function (4.39) with the stability constraints derived in [185], the occurrence probabilities of the two sending intervals are obtained

$$p^d_{40\mathrm{Hz}} = 48\%, \quad p^d_{80\mathrm{Hz}} = 52\%.$$

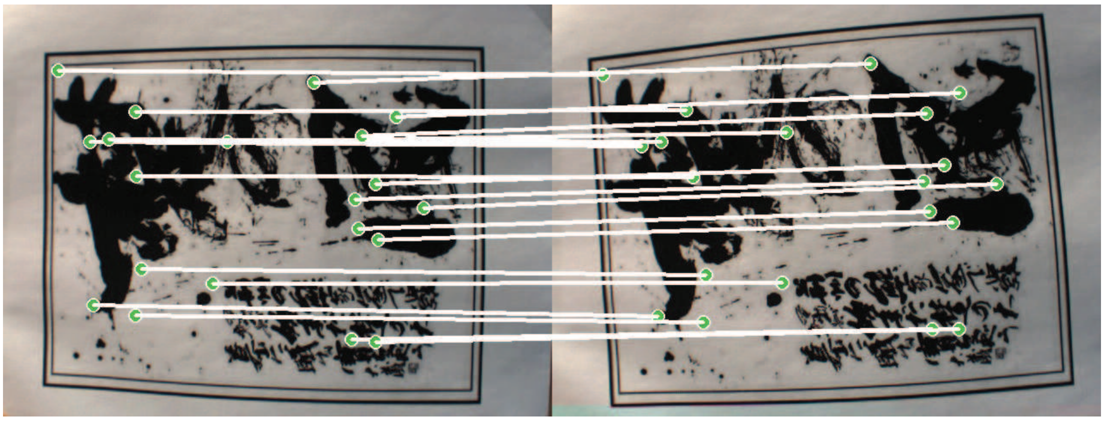


Figure 4.13: Image processing results (feature matching). Left side: reference image; right side: current image. Green point: SURF feature; white line: matched feature pair; number of matched feature pair: 20; image processing time: 18.3 ms.

The associated stabilizing state feedback gains for each joint with $\tau \leq s_1$ are

$$K_{p1} = [120, 120, 60, 60, 30, 30, 30], \quad K_{d1} = [2, 2, 2, 2, 1, 1, 1].$$

The gains for $\tau > s_1$ are

$$K_{p2} = [96, 96, 48, 48, 24, 24, 24], \quad K_{d2} = [2, 2, 2, 2, 1, 1, 1].$$

The tracking error threshold $r_1 = 0.01\,\text{m}$ is heuristically selected.

In order to compare the control performance and the network load, a standard NVSC system without sending rate scheduling is implemented. Both PC$_1$ and PC$_2$ run at 40 Hz, resulting in high communication network load. The occurrence probabilities of the sending intervals become

$$p^d_{40\text{Hz}} = 0\%, \quad p^d_{80\text{Hz}} = 100\%.$$

The overall time delay is categorized only into one interval. The controller with gains

$$K_{p1} = [120, 120, 60, 60, 30, 30, 30], \quad K_{d1} = [2, 2, 2, 2, 1, 1, 1],$$

is applied.

Sending Rate Scheduling According to the scheduling policy described in this section, if at time instant t_k the tracking error $|\mathbf{X}| > r_1$ and $p^d_{80\text{Hz}}(t_k) < p^d_{80\text{Hz}}$, the ncRTP steaming server sends the images at 80 Hz to PC$_1$ and PC$_2$. Otherwise, it sends the images at 40 Hz to only one PC. With this approach, the sampling rate of the feedback is bounded by [40 Hz, 80 Hz]. The operating frequency of different component in NVSC systems is listed in Tab. 4.3.

Experimental Results A sinusoidal function, which has an amplitude of 0.15 m and a frequency of 1 rad/s, serves as the desired trajectory $\mathbf{X_d}$ for the object moving in the $x-y$ plane in the Cartesian space. The tracking error $\mathbf{X}$ between the object motion and camera motion is discussed for performance evaluation.

Table 4.3: Frequencies of components in NVSC system.

Camera	PC_1	PC_2	Visual feedback	Controller
400 Hz	[0, 40Hz]	[0, 40Hz]	[40Hz, 80Hz]	1000 Hz

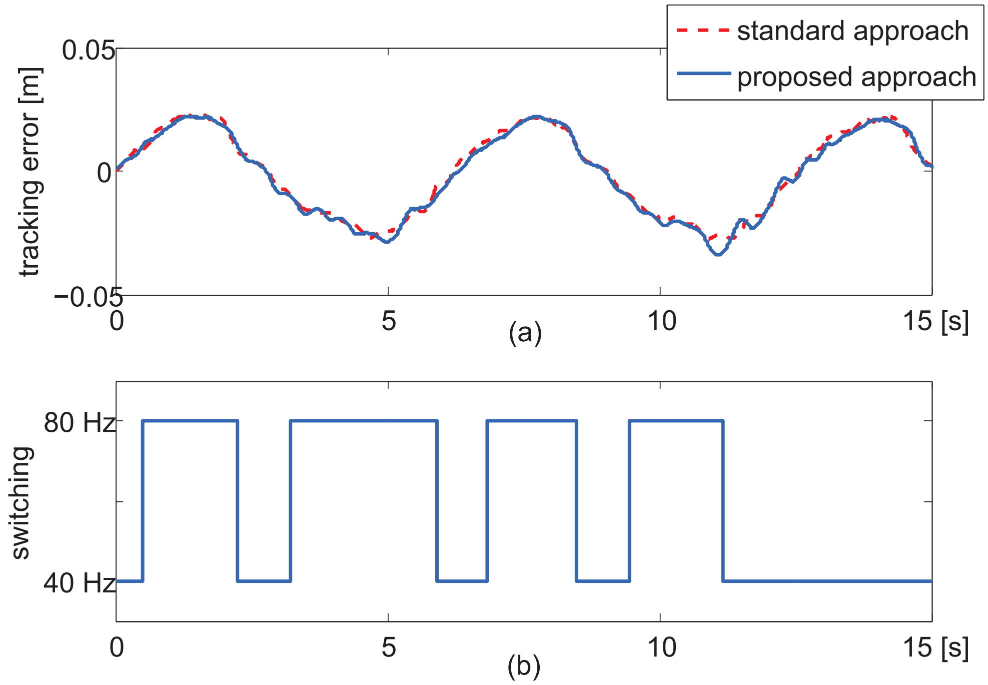


Figure 4.14: Tracking results. a): the control error evolution of the proposed approach with data scheduling (solid line) and the standard approach without data scheduling (dash line). b): Sending rate switching between 80 Hz and 40 Hz for the proposed approach in the experiment. 80 Hz: 52%, 40 Hz: 48%.

The results are shown in Fig. 4.14, and the distribution of delays behaved as desired. As shown in Fig. 4.14(a), a comparable control performance is achieved with and without sending rate scheduling. The proposed approach with sending rate scheduling has a mean tracking error $|\mathbf{X}|_{\text{mean}} = 1.73\,\text{cm}$, which is similar to the mean tracking error of the high sampling rate design approach ($|\mathbf{X}|_{\text{mean}} = 1.67\,\text{cm}$), see Tab. 4.4. However, the communication network load is decreased by 24% compared to the high sampling rate design approach.

The experimental results demonstrate a comparable control performance of the proposed approach with lower communication network load than that of the non-scheduling approach.

4.4 Discussion

The conventional visual servo control suffers from the shorting coming of low-sampling-rate visual feedback due to long image processing delay. By utilizing the networked computation resources parallel image processing is realized, which increases the sampling rate of the visual feedback. However, the resulting stochastic NVSC system with time-varying

Table 4.4: Control performance and communication network load.

	$\lvert\mathbf{X}\rvert_{\text{mean}}$ [cm]	$\lvert\mathbf{X}\rvert_{\text{var}}$ [cm^2]	Communication network load [Mbps]
with scheduling	1.73 (+4%)	0.09	146 (-24%)
high sampling rate	1.67	0.018	192

feedback time delay challenges the stability analysis and controller design. Distributing images over the communication network results in high network load, which leads to the consideration of the cost-performance trade-off.

This chapter has presented a novel analysis and design approach for NVSC systems with cloud computing and sending rate scheduling. In order to improve the control performance, a novel switching control scheme is proposed accounting time-varying time delay. The closed-loop system is modeled as a continuous system with time-varying time delay considering random computation and transmission time delays as well as sampling intervals. The overall time delay is categorized into small delay intervals, and the properties of the time delays are analyzed for selecting the boundaries of the intervals. Mean exponential stability is proven based on occurrence probabilities of delays.

In order to jointly consider the network load due to image transmission and the control performance, a cost function incorporating the control cost and the network usage has been formulated for sending rate scheduling design. Two approaches of sending rate scheduling at the streaming server in NVSC systems have been investigated. Within the first approach, the sending rate scheduling is associated with the probability distributions of different sending interval, which are obtained by optimizing the cost function and meanwhile satisfying the stability constraints. The sending rate at each time instant is randomly selected with the management of the traffic entry in percentage. Within the second approach, the tracking error has been additionally considered in sending rate scheduling in particular for 3D tracking system based on NVSC. The sending rate is adaptively switched according to the current tracking error. Larger tracking error results in higher sending rate, and vice versa. Benefiting from the scheduling mechanism, the network load is effectively reduced; meanwhile, the control performance is guaranteed.

The benefits of the proposed approaches introduced in this chapter are summarized below:

- Better control performance is achieved by the delay-dependent switching control law.
- Control performance and usage of network resources is balanced through sending rate scheduling. The control performance is guaranteed, while the network resources are efficiently utilized.

In this chapter, innovative ideas for control design and cost-performance trade-off have been proposed with promising results. Networked parallel computation aiming at a high-speed visual feedback is effectively utilized to achieve a better control performance. This networked visual servo control concept opens up an important new horizon of research and further studies in vision-based control, in particular in high-speed visual servo control. In this thesis, the tracking error has been considered in the design of sending rate scheduling.

However, the high sending rate even within a short period increases the risks of network congestion and packet loss, which are omitted to some extent in this thesis. Another aspect not yet considered is the non-linear control problem for standard visual servo control system, which has been linearized in this thesis for the ease of system analysis and control design. These aspects are subject to future research directions.

5 Insect-Inspired Vision for Robotics

In the proceeding chapter, high-speed vision systems have been realized through distributed computation over a communication network. The stability problem due to the feedback delay in closed loop vision-based control systems has been investigated. This chapter focuses on high-speed visual servo control from both a control engineering perspective and a neurobiological perceptive enlightened by the robust and high-performance solutions from the neurobiology.

In conventional research visual perception consists primarily of two steps, namely feature detection and feature matching. Generally, these two steps are time consuming when robust image feature detection is required. The large latency caused by image processing deteriorates the overall control performance, and may lead to instability of the system. In order to overcome these limitations in conventional approach, biologically inspired model, in particular the Reichardt-model which has simple architecture and is robust to noise, motivates great interest in the biologically inspired vision system for robot control. The Reichardt-model have been adopted in some works as a qualitative motion detector with constraints on the input stimuli to the vision system, i.e. only allowing sinusoidal grating as input in order to achieve a reliable motion detection. Works considering Reichardt-model as a *quantitative* motion detector in an *unconstructed* environment are not known to the best of the author's knowledge.

The innovation of this chapter is applying a particular insect-inspired motion detector to conventional visual servo control. One goal of this chapter is to improve the accuracy of velocity estimation with the insect-inspired motion detector. The other goal is to investigate the stability of high-speed visual servo control systems with large feedback gains and delays. Contributions are an accurate velocity estimation algorithm based on the analysis of image statistics, and an increase of stability margin for closed loop vision-based control. Key challenges are the analysis of the intrinsic characteristics of the insect-inspired motion detector, the algorithm design for velocity estimation, and the formulation of the insect-inspired visual servo control.

The remainder of this chapter is organized as follows: The insect-inspired motion detector Reichardt-model and its various are introduced and analyzed in Section 5.1. In Section 5.2, the basic characteristics of the Reichardt-model are introduced. The responses of Reichardt-model to sinusoidal signals and natural image sequences are analyzed. An real-time LUT-based motion estimation algorithm is proposed. Simulations and experiments on a 7-DOF robotarm are conducted to validate the proposed approach. In Section 5.3, the stability problem of visual servoing system is investigated. The special response-velocity relationship of the insect-inspired motion detector is utilized to stabilize the closed loop system with high feedback gains and delays. The proposed approach is evaluated by simulation and experiments on 1-DOF linear motor module.

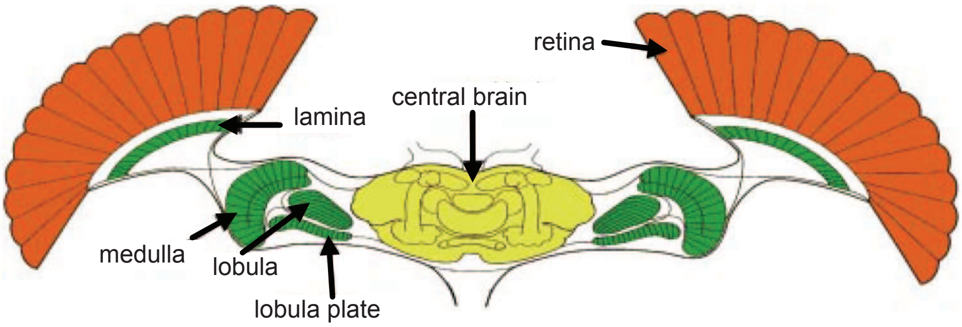


Figure 5.1: The visual system of the fly adapted from [15].

5.1 Background on Biological Motion Detection

In order to applying the insect inspired motion detector to robot control, the vision system of flies is firstly introduced in this section. As found by the neuroscientists Reichardt motion detector is involved in motion perception in a fly's brain [151]. Since the raw Reichardt-model has its limitations in reflecting insects' visual information processing, modified Reichardt-models are often proposed in other research works. In this section, the raw Reichardt-model and its variants are discussed.

5.1.1 Signal Processing in Fly's Brain

The processing of visual motion in fly's brain starts in its compound eyes, each of which consists of 3000-4000 ommatidia ("little eyes") processing their own lens and sets of photoreceptors. The compound eye has a resolution of about 1.5^o, which is equivalent to an image with the resolution of 60×60 pixels (depending on the species) [159]. The fly's visual system has three successive layers of neuropile called lamina, medulla and lobula complex, as shown in Fig.5.1. The lobula complex is split into two parts, the lobula and the lobula plate, and it receives input from medulla elements in parallel. Each column of these layers is formed by a stereotyped set of neurons that are repeated throughout the layer. The response properties of most columnar elements are largely unknown due to their small size compared with the lamina neurons [15, 195–197]. In the lobula plate, a group of visual interneurons with a large size (diameter up to 10 μm) is found, called lobula plate tangential cells (LPTCs), see Fig. 5.2. There are about 60 LPTCs exiting on each hemisphere and they are sensitive to visual motion in specific directions, e.g. they are excited by the motion in one direction called preferred direction (PD), and inhibited by the motion along the opposite direction called nulled direction (ND).

The visual motion information is processed in the fly's brain in a local and hierarchical manner [160]. With the pattern projected onto the retina, photoreceptors in the retina are adapted to the ambient light level and the signal deviations from this level. The output of the photoreceptors is then passed to the lamina layer to perform a temporal bandpass filter of the receptor signals without retaining any information about the background intensity. The local measurements of motion, e.g. between adjacent photoreceptors, are computed in the next layer of medulla cells. However, due to the small size of medulla cells, direct evidence about this processing stage is difficult to record. After local motion estimation, these local, direction-selective motion estimates are then integrated by huge tangential

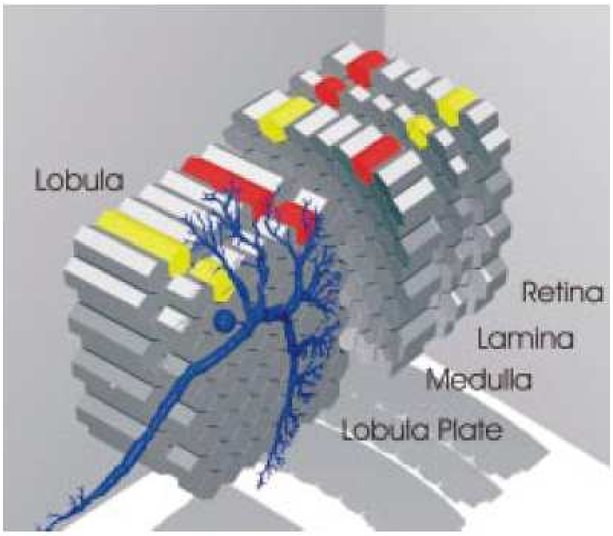


Figure 5.2: Lobula plate tangential cell [15].

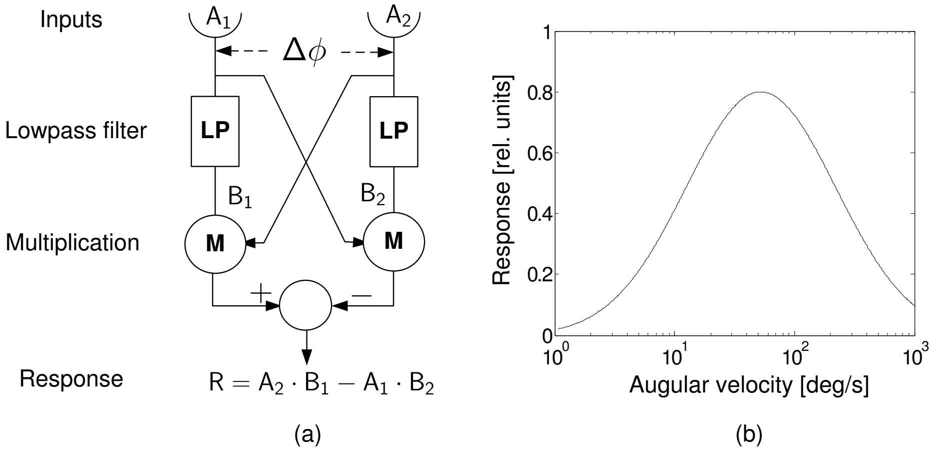


Figure 5.3: Basic Reichardt-model (a) and the typical response-velocity curve (b). $\Delta\phi$: distance between input arms; A_1, A_2: input stimuli; B_1, B_2: delay signal after low-pass filtering; R: output.

cells in the lobular plate [198]. Output of these cells provides information about the translational motion along and rotational motion about different axes in the visual field. This information controls the fly's motor reflexes during flight. Responses of these cells can be modeled by spatial integration of the outputs of elementary motion detectors (EMDs).

5.1.2 Elementary Motion Detector and its Variants

Raw Reichardt-model

The well-known model of motion detector inspired by biological systems was proposed by Reichardt and Hassentein in 1956 [199]. The model is called Reichardt-model and is shown in Fig. 5.3(a). A_1 and A_2 are two photoreceptors and they are temporally delayed by low-pass filters LP, respectively. With $A_1(t)$ and $A_2(t)$ representing the input signals at the left and right input channels, and $B_1(t)$ and $B_2(t)$ representing the corresponding

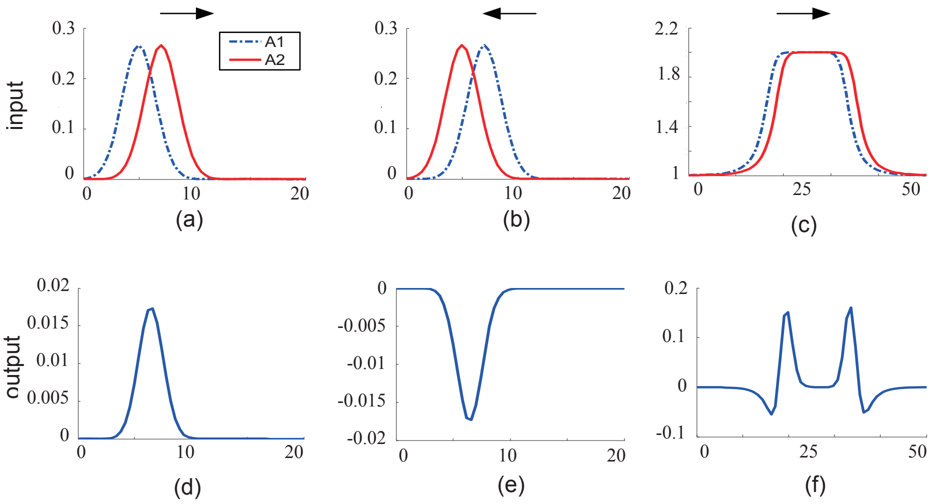


Figure 5.4: Response of the Reichardt detector to a moving peak and to a moving pulse [200]. a) a peak moving to right; b) a peak moving to left; c) a pulse moving to right; d) response to input signal in a); e) response to input signal in b); f) response to input signal in c).

filtered signals, one obtains the response $R(t)$ of the Reichardt-model

$$R(t) = A_2(t)B_1(t) - A_1(t)B_2(t). \tag{5.1}$$

The detector generates a direction sensitive response because of the subtraction between the two symmetric detector halves, see Fig. 5.4 for an example. For the input signal moving right as in Fig. 5.4(a), the response of Reichardt-model is positive in Fig. 5.4(d), while for the input signal moving left as in Fig. 5.4(b) the response becomes negative in Fig. 5.4(e). The velocity dependence of the response of the Reichardt-model is illustrated in Fig.5.3(b). The model response initially increases with increasing velocity, and then decreases after reaching a maximum. It typically presents a bell-shaped response-velocity curve. This specific velocity-response dependency leads to the conclusion, that the gain of the motion detection system in the fly's optomotor pathway is not constant, but becoming smaller at higher velocities [201].

However, the sign of the Reichardt response is not always consistent with the direction of the motion. For example, when the input is a pulse signal as in Fig. 5.4(c), the sign of the response does not directly indicate the motion direction since both positive and negative values appear in the response as shown in Fig. 5.4(f) [202]. The relationship between the motion direction and the sign of the EMD response is summarized in Tab. 5.1. In order to overcome the ambiguity in motion direction estimation with the raw Reichardt-model, variants of EMD have been proposed in known works.

Table 5.1: The relationship between the motion direction and the sign of the EMD response.

Input signal	Motion direction	EMD response
peak	move right	positive
	move left	negative
pulse	move right	positive + negative (ambiguity)

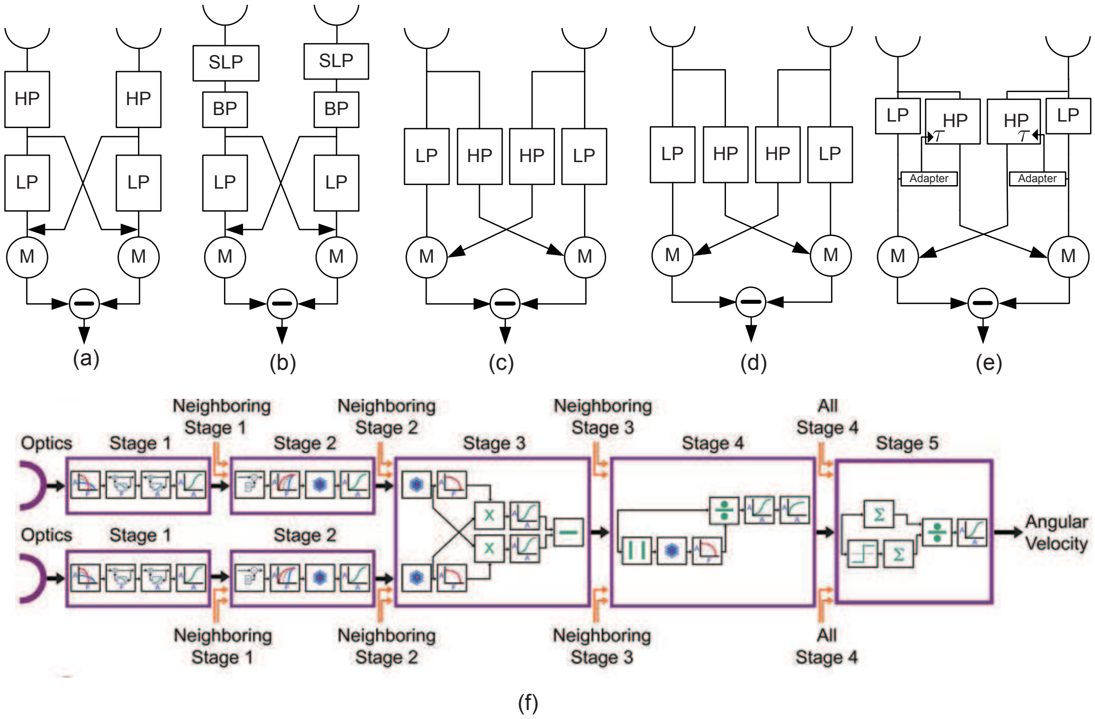


Figure 5.5: Variants of Reichardt-model adapted from [17,155,205,207–209]: (a) EMD with high-pass filter (HP) in both input arms; (b) EMD with spatial low-pass filter (SLP) and temporal bandpass filter (BF); (c) EMD with HP in the cross arms; (d) EMD with subtractions of mean luminance $\bar{I}$ in the input arms and saturation statistics in both branches; (e): EMD with adapter; (f) Fully elaborated EMD, see details in [209].

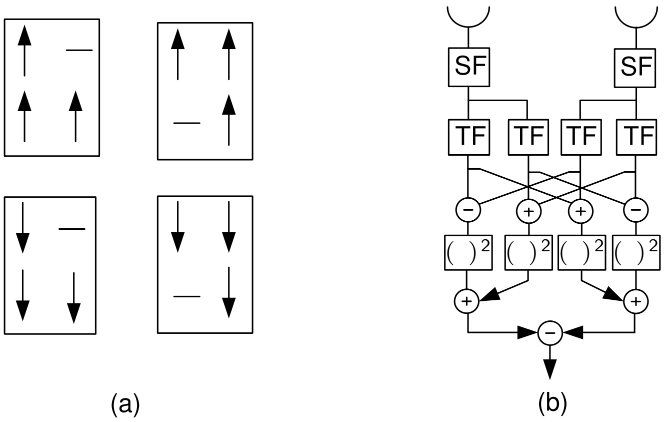
Variants of EMD

Since Reichardt originally introduced the EMD model in the early 1960's, quite a few research work has been done in a wide range of research communities ranging from neurology to psychophysics to show how much appropriately the model can reflect insects' visual information processing systems [203, 204] as well as to propose modified models based on the raw model [156, 205, 206]. In Fig. 5.5 some elaborated models are presented.

In general, there are three methodologies that are used to revise the Reichardt-model:

- *Add preprocessing blocks*: Preprocessing units such as spatial and temporal filters are added in both input signal channels in some research works, e.g. adding a temporal high-pass filter in the input channel to eliminate background luminance, as shown in Fig.5.5(a) [155]. This is necessary if the detector input channels contain saturation nonlinearities. Another example of the elaborated Reichardt-model with preprocessing units is shown in Fig. 5.5(b), which raises the peak response velocity by adding spatial and temporal filtering in the input paths [17]. In [200], a logarithmic transformation is applied to the input arm in order to reduce the sensitivity to lighting conditions. It is recommended by Dror to model the contrast by using a hyperbolic tangent function before the multiplication operation [210].

- *Add components in the cross arm*: In [207] high-pass filters are added on the cross arms as shown in Fig. 5.5(c). The detector with the filter in the cross arms shows a



(a) (b)

Figure 5.6: Template model adapted from [17] for the detection of motion to the right (a) and energy model adapted from [212] (b). The variance of the light intensity is indicated by ↑ (increase), ↓ (decrease), – (no change). SF: spatial filter; TF: temporal filter;

better impulse response performance. The nonlinearity which is used to mimic the saturation characteristics is added to the cross arm and inserted after the low-pass filter, as shown in Fig.5.5(d) [208].

- *Add an adapter*: In order to reduce the motion estimation error, an adapter is added to dynamically adjust the time constant of the filters in [205]. This model is much more complicated than the previous versions and is proved to be a plausible structure underlying the adaptive response properties in the fly's motion vision. In [165] a parameter configuration algorithm is proposed by optimizing the EMD's parameter $\Delta\phi$ (spatial separation between receptors) and τ (time constant of low-pass filter LP) in real time so as to obtain the maximum absolute values of the EMD responses.

A latest and more complex model is shown in Fig. 5.5(e), which is proposed by Brinkworth in 2009. Multiple levels of non-linear dynamic adaptive components are introduced into the model, allowing a more reliable motion detection across different images. There are also some detectors different from the basic Reichardt-model and its alternatives. One example is the template model [156] proposed by Horridge which simulates the function of the small field motion detection neurons in the medulla, shown in Fig. 5.6(a). Another example is the spatiotemporal energy model [211], see Fig. 5.6(b). In this model, the first stage consists of linear filters that are oriented in space-time and tuned in spatial frequency. The outputs of the quadrature pairs of such filters are squared and summed to give a measure of motion energy. It is noteworthy that energy models are closely related to the Reichardt-model and in some cases are formally identical.

It has to be mentioned that, most motion detector and its variants are proposed from the perspective of neurobiology, meaning that the model is revised by comparing the simulation results with the intracellular recording from the interneurons in fly's brain. It is less considered to improve the model from application perspective.

5.2 High-speed Visual Servoing with EMD

In the previous section, the structures of the raw Reichardt-model and its variants have been introduced. The vision system of fly with the involvement of EMD, especially elaborated EMD, enlightens engineers with the design of insect-inspired motion-sensitive sensors, which can overcome the potential problems of conventional machine vision algorithms, e.g. high complexity and computation cost. In this section, innovative insect-inspired approaches, especially a high-speed vision system based on EMD for robot control, are investigated. The Reichardt-model is well-known as an insect-inspired motion detector. However, it often works as a qualitative motion-sensitive sensor rather than a quantitative velocity estimator. In order to enable it as a qualified velocity estimator, an elaborated Reichardt-model is analyzed and applied in this work. A velocity estimation algorithm based on the analysis of spatial power spectrum is proposed and validated in a closed loop vision-based motion control system.

5.2.1 Problem Definition

The Rechardt-model possesses a simple structure and is robust to noise. However, the performance of velocity estimation with the Reichardt-model is not well addressed in known works. The challenge of applying Reichardt-model as a velocity sensor is that its response relies not only on the image velocity, but also on the pattern statistics such as image brightness/contrast and spatial power spectrum density [213–215]. Recent work has shown that with the natural scene as input signal Reichardt-model with additional pre/after-mechanisms, e.g. pre-filters and output integration, is more accurate in velocity estimation [209, 215]. A latest Reichardt-model with complicated structure has been tested for optic flow coding with input stimuli of natural scenes [209]. However, most models applied in these studies have not been tested with real-time image sequences, and the reliability has not been evaluated in closed loop robot control. The input images used in these works are virtually obtained by shifting/rotating a pre-captured image, which leads to similar image statistics among all of the inputs.

In this work, real-time images are taken as input signal, which has varying scenes captured by a rotating camera. Therefore, the varying image statistics have to be considered in velocity estimation. The response of the Reichardt-model to normal scenes is analyzed, and the utilization of the Reichardt-model in a robot control scenario in an unconstructed environment is investigated in this work.

The remainder of Section 5.2 is divided into two parts: In the first part a reliable velocity estimation method with Reichardt-model is proposed. Based on the power spectrum analysis, the response of Reichardt-model to natural image sequence and the design of real-time velocity-response LUT are introduced in Section 5.2.2. In the second part, several steps are proposed to optimize Reichardt-model as accurate velocity estimator are presented in Section 5.2.3, such as tuning of camera view angle and adding pre-filters.

5.2.2 Characteristics of EMD

Though the Reichardt-model or the so called elementary motion detector (EMD) is exploited to be a motion sensitive detector in fly's vision system, it is non-ideal and not

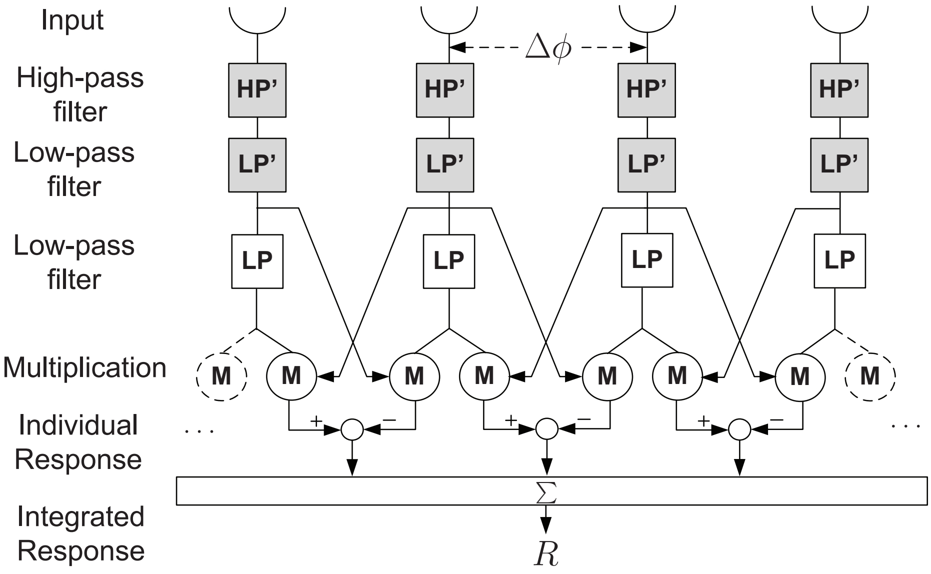


Figure 5.7: An array of elaborated EMDs. HP', LP': addtional temporal high-pass and low-pass filters compared to the raw EMD architecture. R: spatially integrated response of the EMD array.

applicable for high-performance robot control due to the low accuracy of velocity estimation. As mentioned in [216], large standard deviation of the EMD response is observed relative to its mean when it responds to input signals with constant image velocity. In order to reduce the relative error, a reliable EMD model with additional mechanisms, such as pre-filtering of the input signal, needs to be designed. Here, instead of enabling EMD as an insect neuron (e.g. as HS: horizontal-sensitive motion detection neurons) by extending it with a lot of additional components, the relationship between the current image statistics and the EMD response is established from the mathematical point of view. The power spectrum of the input signal is analyzed. An elaborated EMD is proposed in this work, as shown in Fig 5.7. A temporal high-pass filter HP' and a temporal low-pass filter LP' are added on each input path. The role of the two pre-filters will be discussed in Section 5.2.3. Similar to other works, a spatial integration is followed to eliminate the oscillatory components in the response.

Since the image captured by insects varies as a function of time and space, a series of sinusoidal signals with different frequencies and phases is resulted after the Fourier decomposition of the image. Before analyzing the response of EMD to the image sequence, its responses to single sinusoid and to the sum of sinusoids are introduced in the following part.

Remark 6 In this study, the Reichardt-model is utilized for 1-D motion control. An array of EMDs operates horizontally on an image, see Fig. 5.7. The input signals of an EMD are the illumination values of two pixels in the same row. In order to enable it for 3-D motion estimation, more complex patterns (or the so called receptive field [236]) should be designed, which detect different motions by placing EMDs in appropriate ways. However, extracting 3-D motion from these EMD responses is still an open problem, and will be subject to future work.

EMD Response to Single Sinusoid

In this section, the response of basic EMD to single sinusoidal input is studied. Assume the input signal of the EMD is a sinusoid signal $I(t)$ moving with velocity v:

$$I(t) = C\cos(2\pi f v t + \theta_0), \tag{5.2}$$

where C is the amplitude, θ_0 is the phase, and f is the spatial frequency. Then, the temporal frequency of the input signal is $f_t = fv$. Consider that the first-order temporal low-pass filter LP in Fig. 5.3 has a transfer function:

$$G(f_t) = A(f_t)e^{-i\theta(f_t)}, \tag{5.3}$$

where

$$A(f_t) = \frac{1}{\sqrt{1 + (2\pi f_t \tau_L)^2}}, \quad \theta(f_t) = \arctan(2\pi f_t \tau_L),$$

and τ_L is the time constant. With the angular distance between the input arms equals to $\Delta\phi$, the EMD response to the sinusoidal signal in (5.2) becomes [208]

$$R(t) = C^2 \frac{2\pi\tau_L f v}{1 + (2\pi\tau_L f v)^2} \sin(2\pi f \Delta\phi). \tag{5.4}$$

The sign of the EMD response $R(t)$ indicates the direction of the image velocity v. Assume positive v denotes clockwise rotation of the yaw angle. Then, a positive EMD response indicates a clockwise moving direction. In addition, equation (5.4) indicates the response of EMD is a constant and independent of the phase of the input sinusoid, see also Fig. 5.8. With the input signal in Fig. 5.8 (a), the EMD response is given in Fig. 5.8 (c). After about 200 ms, the response in Fig. 5.8 (c) converges to a constant value of 1.1×10^{-3}, indicating a clockwise rotation.

Since the input of EMD is varying luminance, it could not be negative. Thus, a sinusoidal signal with positive mean luminance

$$I(t) = C\cos(2\pi f v t) + K,$$

where $K > 0$, $I(t) >= 0$, is applied as an input signal. The response of the EMD becomes [208, 215]

$$\begin{aligned} R(t) =& C^2 \frac{2\pi\tau_L f v}{1 + (2\pi\tau_L f v)^2} \sin(2\pi f \Delta\phi) + 2CK \sin(\pi f \Delta\phi)[\sin(2\pi f v t - \pi f \Delta\phi) \\ & - A(f_t)\sin(2\pi f v t - \pi f \Delta\phi - \theta(f_t))]. \end{aligned} \tag{5.5}$$

The second term on the right hand side of (5.5) implies a periodic oscillation imposed on the mean response, see also Fig. 5.8. With the input signal in Fig. 5.8 (b), the output of EMD shown in Fig.5.8 (d) is not constant, which leads to ambiguity in velocity estimation. Taking an average response by spatially integrating the responses of an EMD array could solve this problem.

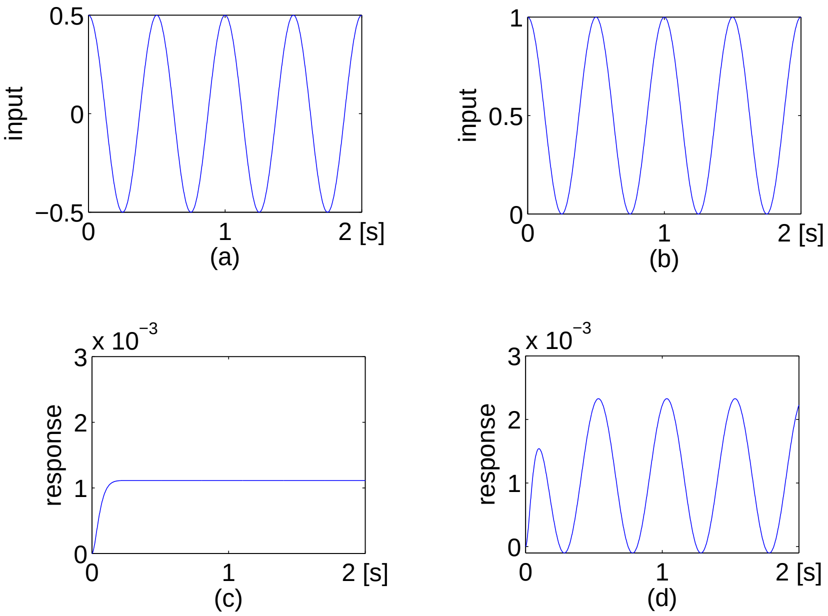


Figure 5.8: Response of EMD to sinusoid. (a): input sinusoid with amplitude 0.5, spatial frequency $f = 0.1$ cycle/deg; (b): input sinusoid with positive mean, K=0.5; (c) EMD response to input signal in (a); (d) EMD response to input signal in (b). $\Delta\phi = 1.1\,^\circ$, $\tau_L = 0.035$ s, and $v = 20$ deg/s for a motion from left to right.

EMD Response to Sinusoids

Since the Fourier decomposition of the image is a series of sinusoidal signals, the response of EMD to the sum of sinusoids is analyzed in this section. Assume there is an input signal consisting of n sinusoids with their corresponding amplitude $C(f_i)$, where $i = 0, 1, 2, \ldots, n$. Due to the non-linear property of EMD, the EMD response differs from the sum of responses to separate sinusoids. The mean response of EMD to the sum of sinusoids is

$$R(t) = \sum_{i=0}^{n} C^2(f_i) \frac{2\pi\tau_L f_i v}{1 + (2\pi\tau_L f_i v)^2} \sin(2\pi f_i \Delta\phi) + O(R), \tag{5.6}$$

where $O(R)$ denotes the error between the actual response and the sum of responses to separate sinusoids. According to (5.6), when $O(R)$ approaches zero, the EMD response is equivalent to the sum of responses to separate sinusoids. Then, a reliable velocity-response relationship can be established depending only on the spatial frequencies f_i, $i = 0, \ldots, n$ of the sinusoidal components in the input signal. Therefore, the objective here is to minimize the difference $O(R)$ between the theoretical prediction R_p and actual response of EMD R_a when using the EMD for accurate velocity estimation. Here, the actual mean response of EMD R_a represents the response after spatial averaging, $R_a = R/N$, where R denotes the measured integrated response of an EMD array in Fig. 5.7, and N denotes the number of EMDs in the array.

5.2.3 Accurate Velocity Estimation with EMD

In this section nature images are used as input to EMD, and the mean EMD response is analyzed for velocity estimation. In addition, the role of camera view angle, spatial resolution, and additional pre-processing modules in the elaborated EMD model are discussed. Finally, velocity estimation algorithm based on real-time LUT is illustrated.

Power Spectrum Density of Natural Scenes

Natural scenes, which are different from place to place and from time to time, are more arbitrary than artificial stimuli in constructed environment (e.g. sinusoidal grating). However, a number of studies [217–219] have shown that the power spectrum of the natural scene possesses a similar distribution despite the diverse scenes in the images. In spite of the arbitrariness of the images, the amplitude of the one-dimensional (1-D) power spectrum falls off roughly by $1/f^{(1+\eta)}$ in log-log coordinates, where f denotes the spatial frequency, and η is a small constant, see Fig. 5.9 for an example.

Although the power spectra of nature images are similar, their actual values alter from image to image. For accurate velocity estimation with EMD, it is necessary to analyze the power spectrum of each input image. The power spectrum of an image is defined as the square of the modulus of the Fourier transform [220]. This work concentrates on a yaw-angle control problem. Therefore, the spacial power spectrum of 1-D row is calculated by

$$P(f_i) = \frac{1}{N^2} \sum_{n=0}^{N-1} (x(nT_s) \exp(\frac{-j2\pi}{N} f_i n))^2, \quad f_i = 0, \ldots, N-1, \tag{5.7}$$

where $x\left(nT_s\right)$ denotes the pixel value, T_s and N denote the spatial sampling interval and the length of the sampling points in one row, respectively.

The 1-D power spectral densities (PSDs) of the images shown in Fig. 5.9 are analyzed. The four images are captured with different surroundings, e.g. the scene in 'classroom' is significantly different from that in 'outdoor'. As shown in Fig. 5.9, the PSDs of the four images differ from each other, while, however, these PSDs all fall off approximately by $1/f^{1.25}$ in log-log coordinates.

EMD Response Prediction

Assume the power spectrum of the input image is known. According to (5.6), the predicted mean response of the EMD becomes

$$R_p = \sum_{i=0}^{N-1} P(f_i) \frac{2\pi\tau_L f_i v}{1 + (2\pi\tau_L f_i v)^2} \sin(2\pi f_i \Delta\phi). \tag{5.8}$$

The relationship between the predicted mean response R_p and the image velocity v is depicted in Fig. 5.10. The input images, which possess PSDs of different slope{-0.9, -1.0, -1.1} shown in Fig. 5.10(a), are shifted with different image velocity. The corresponding mean EMD responses are predicted by (5.8), and the results are shown in Fig. 5.10(b). It is observed that the three normalized response-velocity curves present a similar bell-shaped relationship, except minor differences in the descending part. In order to eliminate the

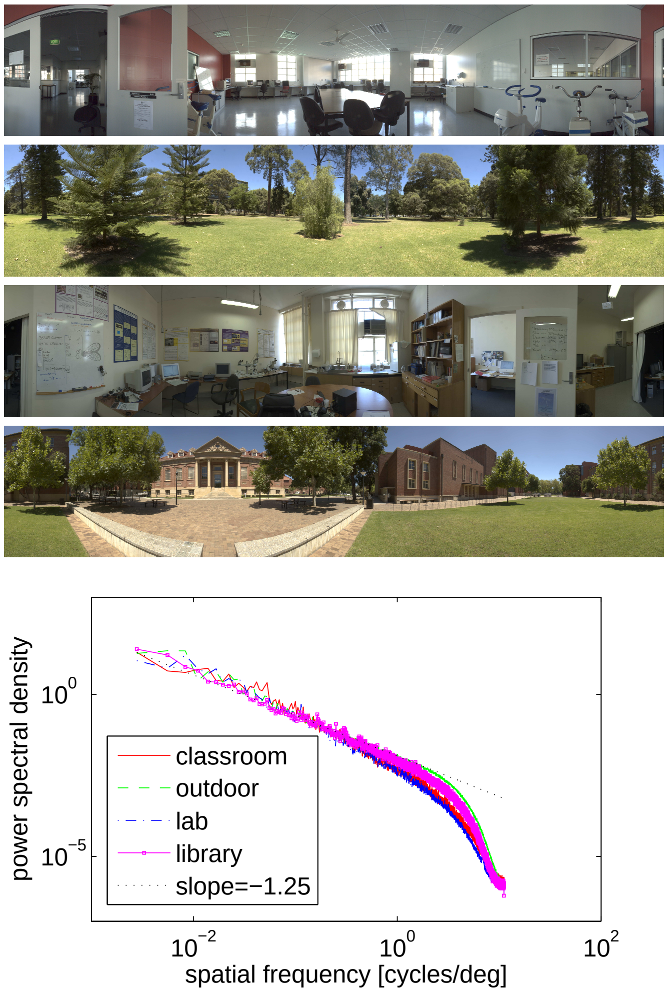


Figure 5.9: Images of different scenes from [209] (captured by a Nikon D-70 digital camera) and their power spectrum densities. Top down: 'classroom'; 'outdoor'; 'lab'; 'library'; horizontal mean power spectral density of images, the dashed line represents power spectra $P(f)$ with $1/f^{1.25}$ in log-log coordinates. A rectangular window with the size of image width is applied for calculating the mean power spectral densities.

ambiguity in velocity estimation, only the ascending part of the curve is utilized for robot control.

In order to estimate the image velocity, the predicted response R_p is compared with the actual response R_a. As pointed out in Section 5.2.2, accurate velocity estimation with EMD relies on the assumption $O(R) \approx 0$. In the following part, several steps are proposed to minimize the difference between the predicted response R_p and the actual response R_a.

Role of Additional Mechanisms

An elaborate EMD model shown in Fig. 5.7 is proposed in this work for velocity estimation. Prior to introducing the functional roles of additional mechanisms, components with different spatial frequencies are discussed with respect to their contributions to the mean response R_p.

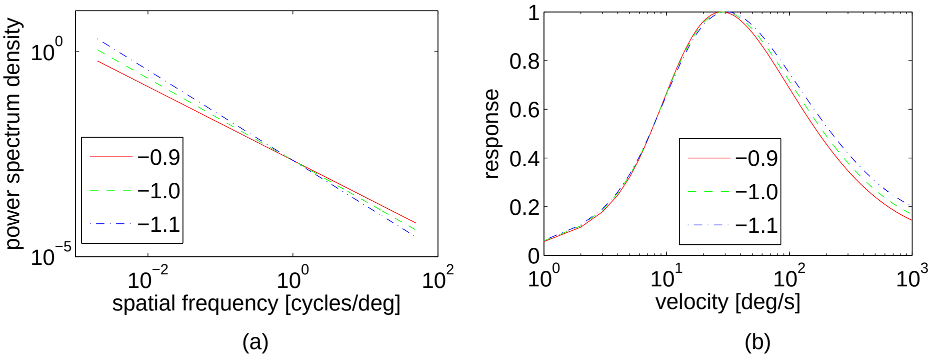


Figure 5.10: Power spectral density of slope {-0.9, -1.0, -1.1} (a) and corresponding normalized response-velocity curves (b).

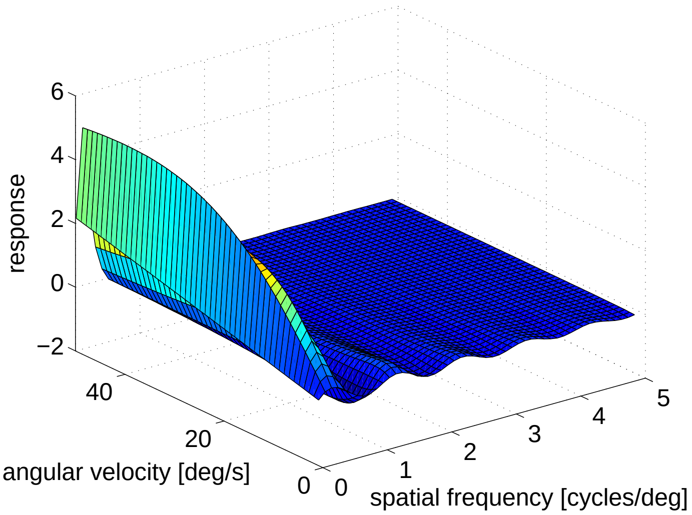


Figure 5.11: The relationship among the predicted mean EMD response, image velocity, and spatial frequency of the input signal. The power spectra of the input signal has a slope of -1.0, with $\eta = 0$.

Contributions of Components with Different Spatial Frequencies Assume the input image possesses a power spectrum of slope -1.0. The EMD response to each sinusoidal component in this image is calculated. The results are shown in Fig. 5.11, which depicts the overall relationship among the predicted response, the image velocity, and the spatial frequency of the input signal. As observed in Fig. 5.11, the response of EMD is not always positive. For example, according to (5.8) the responses to the components with spatial frequency from 0.56 cycle/deg to 1.11 cycle/deg are negative when $\Delta\phi$ equals to $0.9\,^{\circ}$. If these components are dominant in an input image, the mean EMD response may become negative, which is different from the sign of the image velocity v. Accordingly, the velocity estimation becomes incorrect. Therefore, the components, which lead to negative responses when the input signal has positive velocity, are not preferred in velocity estimation.

Moreover, with higher spatial frequency, the response-velocity curve has a smaller peak value at a smaller velocity. Generally speaking, the component of low spatial

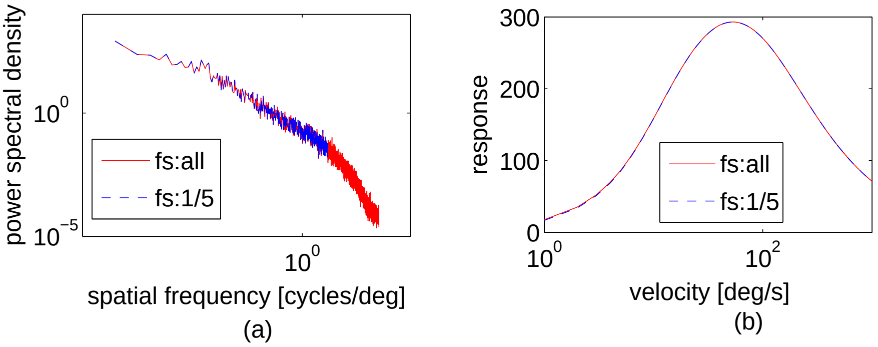


Figure 5.12: Response-velocity curves of input signals with two different spatial frequency ranges. (a): the mean row PSD of image 'classroom' in Fig. 5.9. Red line: the PSD for spatial frequency ranges from 0 to 11.11 cycle/deg; dashed blue line: PSD for spatial frequency from 0 to 2.22 cycle/deg (1/5 of the solid red-line). (b): the correspondent response-velocity curves.

frequency contributes more to the EMD response than that of high spatial frequency. In order to clarify this, the response-velocity curves of the input signal with different range of special frequency are compared in Fig. 5.12. Two ranges of spatial frequency, [0 11.11] cycle/deg (original) and [0 2.22] cycle/deg (fifth of the original) shown in Fig. 5.12(a), are tested. The response-velocity curves are shown in Fig. 5.12(b). The curves indicate that the components of low spatial frequencies play a dominant role in the response-velocity curve. The maximum relative difference $\max(\frac{|R_{\text{all}}-R_{1/5}|}{R_{\text{all}}})$ (R_{all}: response curve to original input signal, $R_{1/5}$: response curve to input with only low spatial frequency) between the two curves is only 3.53%, appearing at $v = 2.0$ deg/s. In consideration of this characteristic, the computation time in application can be reduced by calculating only the responses of the components with low spatial frequencies.

Role of Field of View In machine vision a camera with wide view angle provides large field of view by the price of low spatial resolution. In order to find out the relationship between the width of the field of view (FOV) and the mean EMD response, images (possessing 1000 pixels in one row) with different FOV $\in \{45°, 180°, 360°\}$ are tested in this section. The correspondent responses are shown in Fig. 5.13. Besides, the relative response error e_R defined by

$$e_R = \frac{|R_a - R_p|}{R_a}, \tag{5.9}$$

is utilized to evaluate the difference between R_a and R_p. The mean relative error $\bar{e}_R$ and the variance of the actual mean EMD responses R_a are listed in Tab. 5.2.

It is noticed that the relative mean error is in a small scale for all the three FOVs. However, the image with larger FOV results in a smoother mean response. For example, the variance of R_a by an input image with a FOV of 45° is 100.38, while that by an image with a FOV of 360° is only 28.3 (−71.8%). Since for a constant velocity a response without variation is expected, a camera with large FOV is suggested.

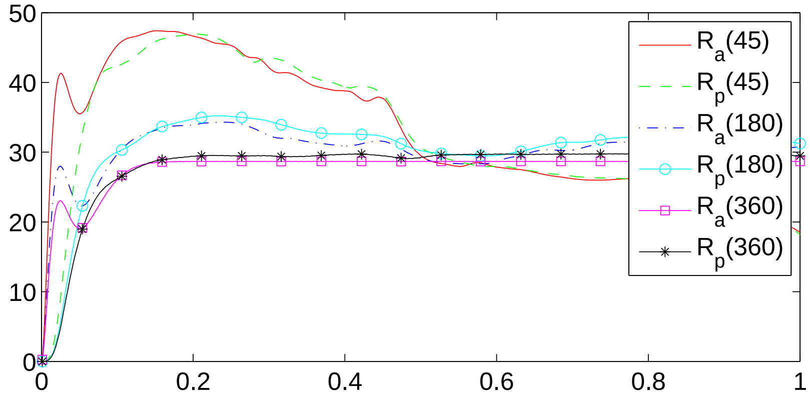


Figure 5.13: The predicted response R_p (dashed line) and actual mean EMD response R_a (solid line) to input images of different FOV: 45°, 180°, 360°. The responses are obtained at the image velocity 45 deg/s with input image 'classroom' in Fig. 5.9.

Table 5.2: Evaluation of the responses. Left: results from the simulation with different FOV, but the same number of pixels in the row (1000 pixels); Right: results from the simulation with different spatial resolution, but the same FOV (180 °).

FOV	mean (e_R)	var (R_a)
45°	0.033	100.38
180°	0.033	33.81
360°	0.022	28.33

Number of pixels in row	mean (e_R)	var (R_a)
500	0.032	33.82
1000	0.033	33.81
4000	0.022	29.96

Remark 7 In order to obtain input image sequence, a virtual camera is simulated which rotates clockwise around yaw axis with an image velocity of 45 deg/s. The image captured by the camera is panoramic (8000 pixels in one row) with different starting position. Since only less than 8000 pixels, e.g. 1000 pixels, are used in the simulation, it requires a sub-sampling on the panoramic image. After sub-sampling on the panoramic images with different starting position, the resulting image differs from frame to frame, leading to varying responses.

Role of Spatial Resolution In order to investigate the relationship between the spatial resolution and the mean EMD response, input images with the same FOV but different number of pixels in one row $\in \{500, 1000, 4000\}$ are tested. The FOV of 180° is adopted in the simulation. The simulation results are shown in Fig. 5.14 and Tab. 5.2. The results indicate that the spatial resolution does not significantly influence the mean EMD response. Combined with the results in the previous section, a wide FOV camera with low spatial resolution is preferred in applications considering the trade-off between the computation cost and the accuracy.

Role of Pre-filters Further investigations are necessary in order to reduce the oscillation in the response and minimize the difference between R_p and R_a. As introduced in Sec-

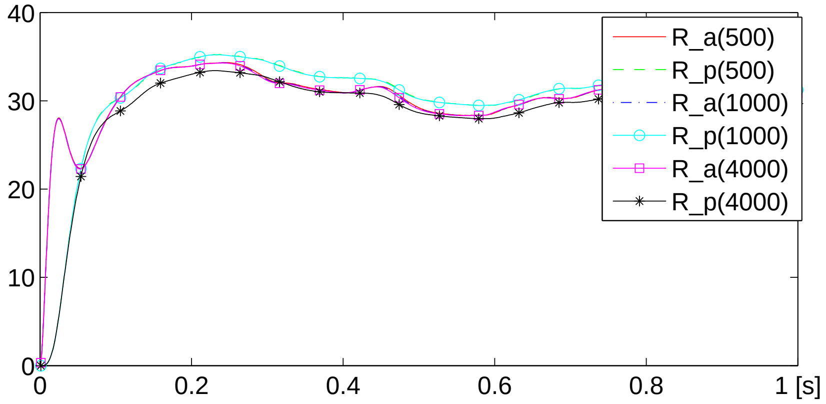


Figure 5.14: The predicted response R_p (dashed line) and actual mean EMD response R_a (solid line) to input images of different spatial resolution: 500, 1000, 4000 pixels in one row. The responses are obtained at the image velocity 45 deg/s with input image 'classroom' in Fig. 5.9.

tion 5.2.2, two pre-filters, temporal high-pass filter and temporal low-pass filter, are added to the basic Reichardt-model, as shown in Fig.5.7.

Temporal High-pass Filter: As pointed out in [221], sufficient spatial integration eliminates the oscillation in the response. Then the response of EMD can be predicted by (5.8). However, it is difficult to meet the requirement of 'sufficient integration' in application. On one hand, spatial integration is dependent on the FOV of the camera. A camera with limited FOV restricts the spatial integration to be insufficient. On the other hand, components of lower spatial frequency contributes more to the response and the impact, however, also more significantly on the fluctuation of the response. Therefore, a high-pass filter is applied to filter out the components of very low spatial frequency. As a result, the variation of the response is reduced and the difference between R_p and R_a becomes smaller.

For example, Fig. 5.15(a) shows the predicted mean response R_p and the actual mean response R_a of the basic EMD model, while Fig. 5.15(b) shows that of the EMD model with the high-pass filter. The relative response error e_R defined by (5.9) is shown in Fig. 5.15(c) and Fig. 5.15(d). The mean relative error $\bar{e}_R$ in Fig. 5.15(c) is about 23.9%, while the one in Fig. 5.15(d) is only 4.15% (-82.6%). The relative error is considerably reduced by adding the high-pass filter.

Temporal Low-pass Filter: It is noticed that for the basic EMD the response-velocity curve might have a local minimum (negative) appearing at very low velocity. An example is shown in Fig. 5.16(b), where the dashed red-line and solid blue-line denote the response-velocity curves obtained with and without the low-pass filter, respectively. On the dashed line a local minimum -0.1 is observed at the velocity $v = 0.8$ deg/s. As explained in Section 5.2.2, with $v = 0.8$ deg/s a positive mean EMD response is expected. Therefore by using the red line the result of velocity estimation becomes incorrect.

By adding a low-pass filter on the input path, the negative responses are eliminated, see the solid line in Fig. 5.16(b). The reason is that by using the low-pass filtering the PSD of the image is altered, see Fig. 5.16(a). The solid line denotes the PSD of the

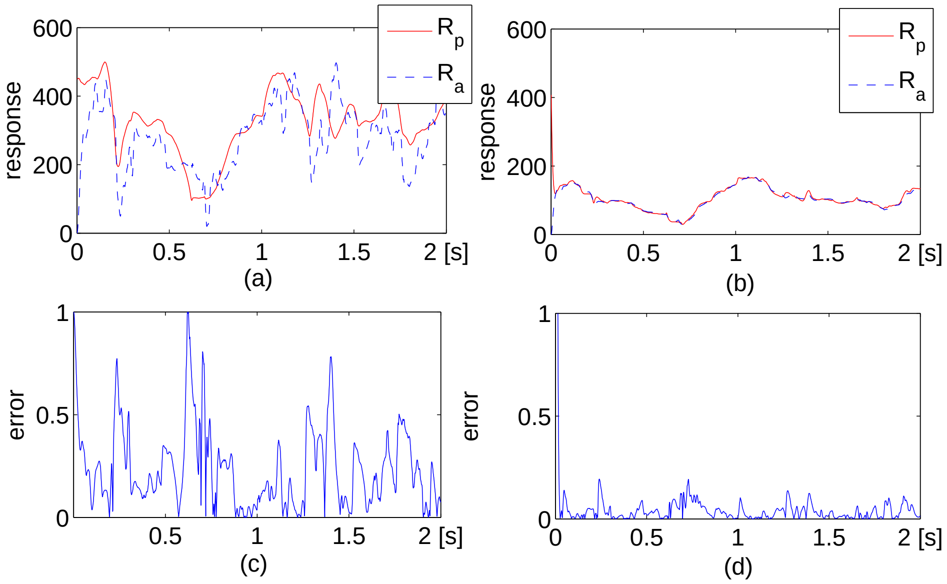


Figure 5.15: Responses of EMD with (right column) / without (left column) high-pass filter. The input image is 'classroom' in Fig. 5.9 with image size 8000×1600 pixels. At each time, an image frame of size 800×27 pixels is captured as input signal. a): predicted response (solid line) R_p and actual output (dashed line) R_a of basic EMD; b): R_p and R_a of EMD with high-pass filter; c): the relative response error of responses in a); d): the relative response error of responses in b). The parameters of EMD are $\tau_L = 0.035\,$s, $\tau_{H'} = 0.02\,$s, $\Delta\phi = 0.9\,^\circ$ and $v = 90\,$deg/s.

image before using the low-pass filter, while the dashed line denotes that after using the low-pass filter. According to (5.8), the negative responses at low velocities result from the components of frequencies $f_i \in [0.56, 1.11]$ cycle/deg when $\Delta\phi$ equals to $0.9°$. As observed in Fig. 5.16(a), in the dashed line the PSDs of spatial frequencies within the range of $[0.56, 1.11]$ cycle/deg are much larger. Besides, as shown in Fig. 5.12 the components of high frequencies contribute little to the EMD response. Therefore, when the components of frequencies within $[0.56, 1.11]$ cycle/deg become dominant in the image, negative responses may be generated, see the dashed line in Fig. 5.16(b). By adding a low-pass filter, the local minimum is successfully eliminated.

Remark 8 It should be pointed out that the negative local minimum appears at very low velocity ($< 1\,$deg/s). For applications without velocity in this scale, the low-pass filtering can be omitted to reduce the computation time. Combining the low-pass filter and the high-pass filter into one filter could also save the computation time.

Algorithm

The discussion above demonstrates the possibility of utilizing the mean EMD response for velocity estimation. In this part, the algorithm for velocity estimation is introduced by establishing a real-time LUT. Though the actual mean EMD response R_a can be predicted by (5.8) based on the PSD analysis and the known velocity v, the velocity cannot be calculated in a closed form. Owning to the non-linearity of (5.8), the velocity v cannot be directly deduced by mathematical means. An alternative is to establish a response-velocity LUT.

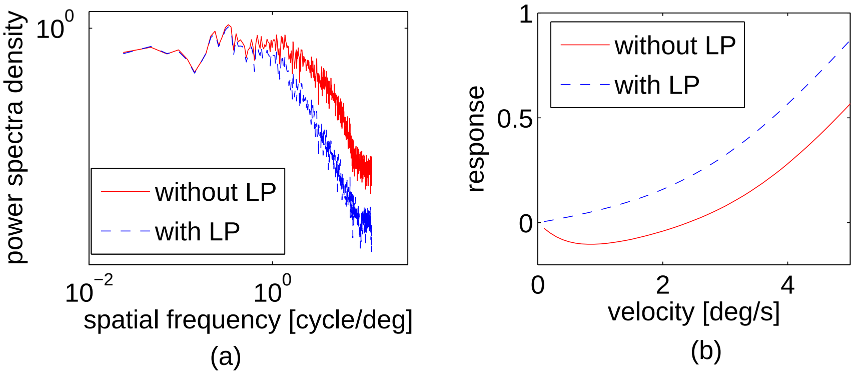


Figure 5.16: Responses of EMD with/without low-pass filter. a): power spectrum of image 'classroom'; solid line: power spectrum of the image before low-pass filter; dashed line: power spectrum of the image after low-pass filter. b): corresponding response-velocity curves. $\tau_{L'} = 0.015\,\text{s}$, other parameters: see Fig. 5.15.

By comparing the actual mean EMD response R_a with the predicted response-velocity LUT, the velocity v is estimated. The algorithm for velocity estimation is illustrated in Fig. 5.17:

- Firstly, the actual mean EMD response R_a is calculated by applying an array of EMDs to the input image;

- Secondly, the PSD of the image after high-pass and low-pass filtering is calculated by (5.7). According to (5.8), a real-time response-velocity LUT is established;

- Finally, by searching R_a in the LUT the estimated image velocity $\hat{v}$ is determined.

Notice the LUT should be updated from frame to frame. Although most natural images have similar PSD slopes, the small difference in the PSD leads to different mean EMD responses. It accordingly results in diverse response-velocity curves. Fig. 5.18 shows some examples of LUT with different input images. Four images, 'img1', 'img2', 'img3' and 'img4', are adopted as input signals, which are sub-segmented from the original image 'classroom' in Fig. 5.9. It is observed in Fig. 5.18, that the PSDs of these images exhibit similar slopes. However, the actual values of these PSDs are different, which thus lead to different response-velocity curves. In order to achieve accurate velocity estimation, a real-time LUT has to be established.

Compared with the existing methods, the advantage of the proposed approach is, that it is applicable in almost all kinds of environments, while the other models/methods are restricted to either simple monotonous artificial environments or natural environments possessing very close PSDs (e.g. [209, 215]). Besides, the accuracy of velocity estimation is improved with the proposed approach, which, for the first time, allows applying the EMD for closed loop motion control in normal environments.

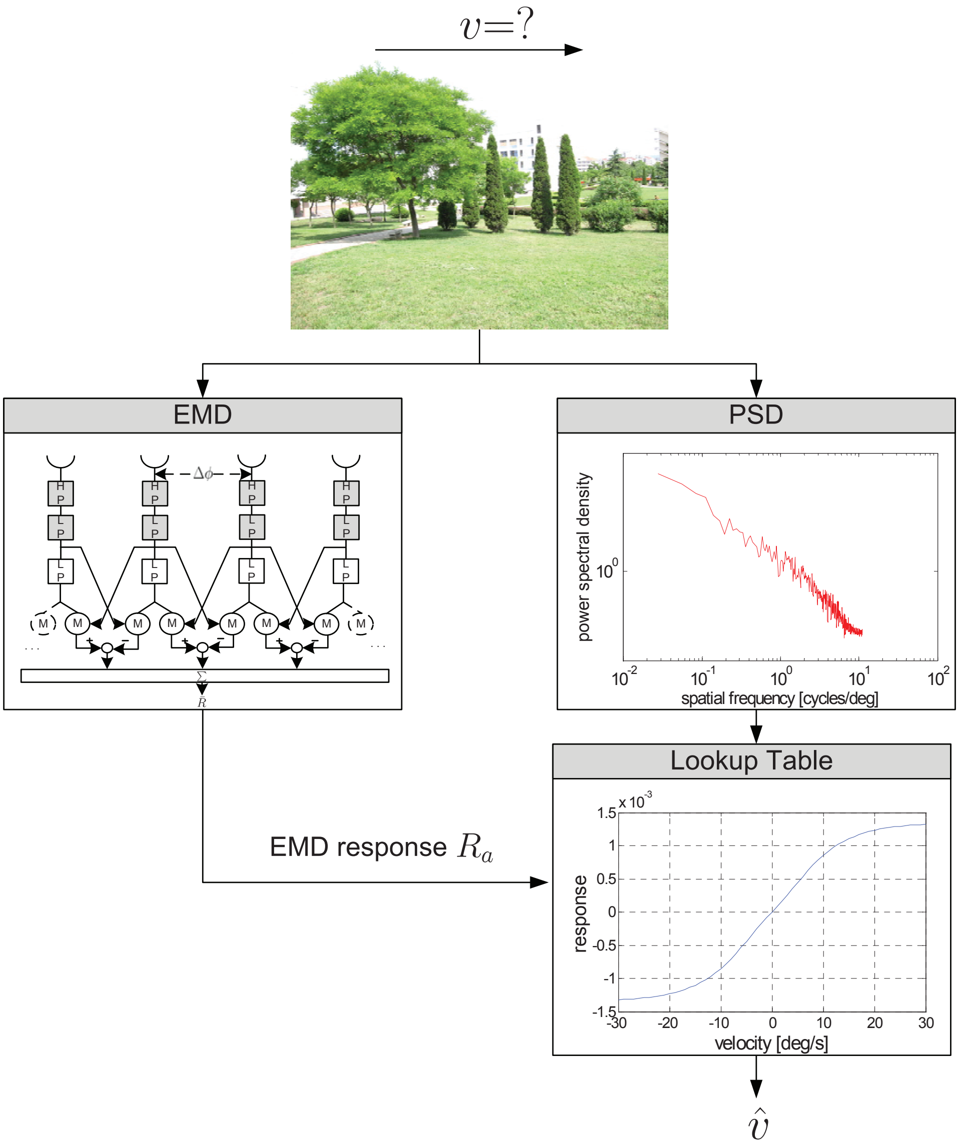


Figure 5.17: Algorithm of applying EMD for velocity estimation based on PSD analysis.

5.2.4 Performance Evaluation

Simulations

In order to verify the performance of the proposed algorithm for velocity estimation, a closed loop system with velocity feedback provided by EMD is designed for velocity control, in which the error is a function of optical flow [222, 223]. The simulation setup and the control performance are introduced below.

Simulation Setup The closed loop system for yaw rate control is shown in Fig. 5.19(a). The desired yaw rate is denoted as $\dot{\theta}_{\mathrm{d}}$. The current yaw rate $\dot{\theta}_{\mathrm{is}}$ is estimated based on the algorithm introduced in Section 5.2.3. The estimated yaw rate is denoted as $\dot{\theta}_{\mathrm{emd}}$. With the feedback $\dot{\theta}_{\mathrm{emd}}$, the angular velocity error $\Delta\dot{\theta} = \dot{\theta}_{\mathrm{d}} - \dot{\theta}_{\mathrm{emd}}$ is then calculated and linked to a PD controller. The output of the controller u is sent to the manipulator for yaw rate

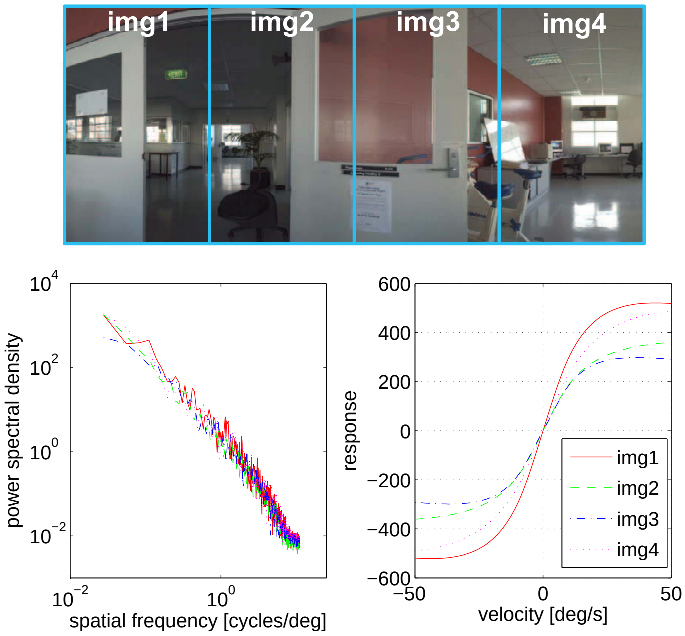


Figure 5.18: Response-velocity curves for different input images, 'img1', 'img2', 'img3' and 'img4', which have similar slope of PSD. The four images are sub-segmented from 'classroom' in Fig. 5.9.

control. The manipulator is simulated by an integrator. The image sequence is captured by a virtual camera rotating clockwise with a yaw rate $\dot{\theta}_{\mathrm{is}} = 45\,\mathrm{deg/s}$. It captures images from the original panoramic image 'classroom' (8000×1600 pixels) in Fig. 5.9, with the size of 2000×8 pixels, see Fig. 5.19 (b) for visualization. The sampling rate of the whole system is set to be 1000 Hz.

In the simulation the resolution of velocity estimation needs to be considered, e.g. for a panoramic image of 8000×1600 pixels the spatial resolution is 0.045 deg/pixel. Running the camera with a frame rate of 1000 cycle/deg, the resolution of image velocity is 45 deg/s. In order to achieve higher velocity resolution, sub-pixel interpolation should be applied. In this simulation, a linear sub-pixel interpolation is exploited to achieve a velocity resolution of 0.1 deg/s. It means between every two adjacent original pixels 449 sub-pixels are linearly interpolated.

The parameters of EMD in the simulation are set as follows: $\tau_L = 0.035\,\mathrm{s}$, $\tau_{H'} = 0.02\,\mathrm{s}$, $\tau_{L'} = 0.015\,\mathrm{s}$ and $\Delta\phi = 0.9\,°$, where $\tau_{H'}$ and $\tau_{L'}$ are the time constants of the preprocessing modules: the temporal high-pass filter and the temporal low-pass filter in the elaborated EMD model. Similar to (5.9), the relative yaw rate error defined by

$$e_{\dot{\theta}} = \frac{|\dot{\theta}_{\mathrm{is}} - \dot{\theta}_{\mathrm{d}}|}{\dot{\theta}_{\mathrm{d}}}, \tag{5.10}$$

is adopted for performance evaluation.

Simulation Results

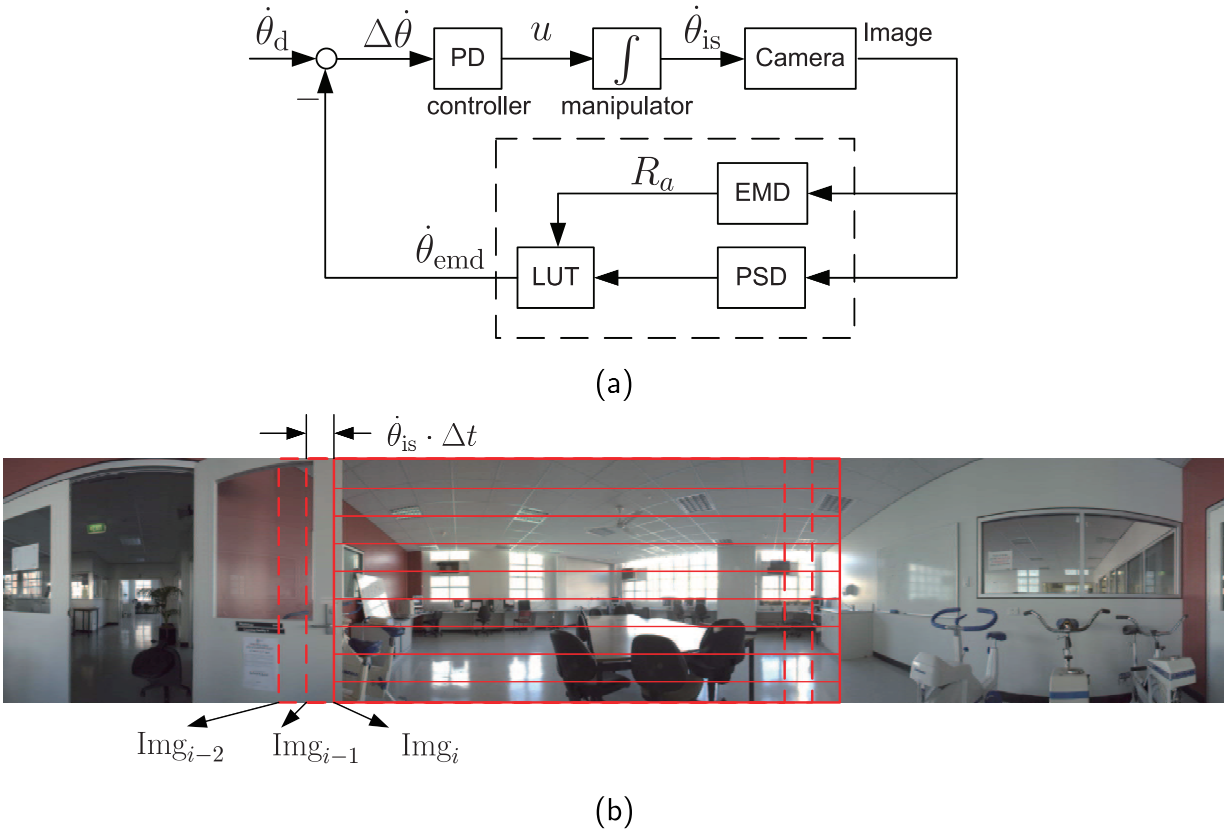


Figure 5.19: Closed loop yaw-rate control system for simulation (a) and Images captured by virtual camera moving with angular velocity $\dot{\theta}_\mathsf{is}$ (b). EMD-based angular velocity estimation is denoted by the dashed rectangular in (a). $\dot{\theta}_d$: desired yaw-rate; $\dot{\theta}_\mathsf{emd}$: estimated yaw-rate; $\dot{\theta}_\mathsf{is}$: actual yaw-rate of the manipulator; $\Delta\dot{\theta}$: control error; PD: controller; $\int$: integrator.

Simulation I: In order to test the performance of the EMD as a velocity estimator for closed loop yaw rate control, a step velocity $\dot{\theta}_\mathrm{d}(t)$ defined by

$$\dot{\theta}_\mathrm{d}(t) = \begin{cases} 5 \text{ deg/s}, & \text{if} \quad t > 0, \\ 0, & \text{if} \quad t = 0, \end{cases}$$

is selected as a reference signal. The results are shown in Fig. 5.20. After an initial delay of about 1 s, the yaw rate of the manipulator converges to the desired value. The relative yaw rate error is shown in Fig. 5.20(b). The mean relative yaw rate error $\bar{e}_{\dot{\theta}}$ during period $t \in [1 \;\; 12]$ s is only 1.25%. The results above demonstrate that reliable velocity estimation is achievable with the insect-inspired EMD, which can be further extended in machine vision for robot control.

Simulation II: The proposed approach is also evaluated in a simulation with a desired motion, which has acceleration and deceleration during the running time. The following reference signal is selected:

$$\dot{\theta}_\mathrm{d}(t) = 3\sin(2\pi 0.1t - \frac{\pi}{2}) + \sin(2\pi 0.5t) + 10, \quad t \geq 0.$$

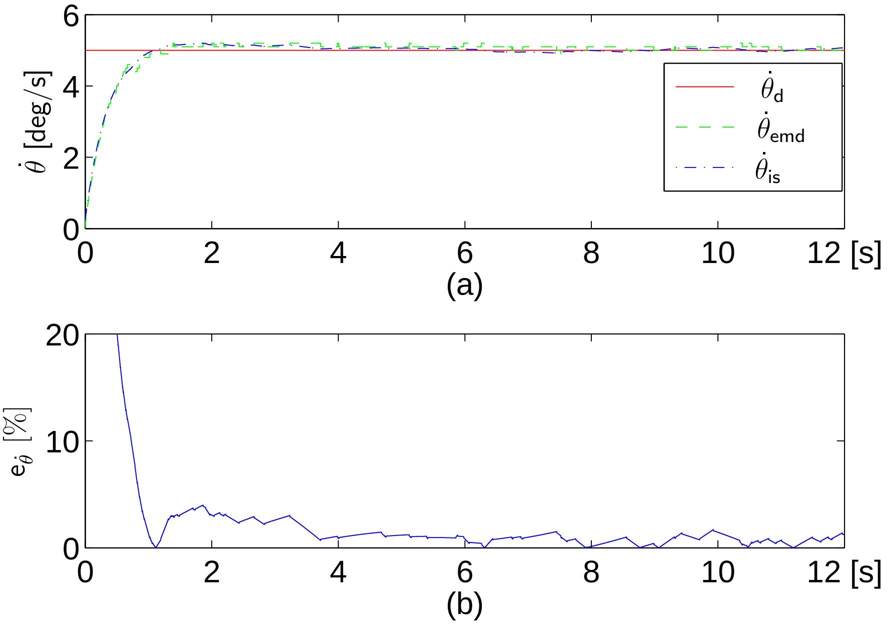


Figure 5.20: Simulation results of using the mean EMD response as velocity estimator in a closed loop yaw rate control. The desired yaw rate is constant, $\dot{\theta}_d(t) = 5\,\text{deg/s}$, $t > 0$. a): yaw rate evolution. Solid red-line: desired yaw rate; dashed green-line: estimated yaw rate based on EMD; dashed-dot blue-line: actual yaw rate of the manipulator; b): relative error evolution according to (5.10).

The results are shown in Fig. 5.21. The velocity of the manipulator follows the desired velocity closely, as seen in Fig. 5.21(a). The relative error defined by (5.10) is shown in Fig. 5.21(b). Similar to the results in Simulation 1, only small relative error is observed after the initial transient response, e.g. the mean relative velocity error is about 5.56% during $t \in [1\;\;12]$ s. Therefore, it is concluded that the proposed method is feasible to be employed as a velocity estimator in robot control system.

Experiments

In this section, the utilization of the EMD as an angular velocity estimator is exploited in real-time robot control. Instead of implementing a yaw rate control system, a vision-based yaw angle control is designed. The reason is that there is unexpected friction on the manipulator, which is difficult to compensate and has large influence on the velocity control. The closed loop yaw angle control system is shown in Fig. 5.22. The control aim is to drive the robot end-effector to follow a designed rotational trajectory θ_d. After obtaining a new image captured by the camera, EMD is applied to estimate the angular velocity $\dot{\theta}_\text{is}$. The estimated angular velocity $\dot{\theta}_\text{emd}$ is followed by an integrator. Thus, the yaw angle θ_emd of the end-effector is obtained and fed back to the controller for computing the control signal u. Benefiting from the simple structure of EMD and the choice of a small size input image, the vision system based on EMD achieves a frequency of 1000 Hz.

Experimental Setup The experimental setup is shown in Fig. 5.23. The end-effector of

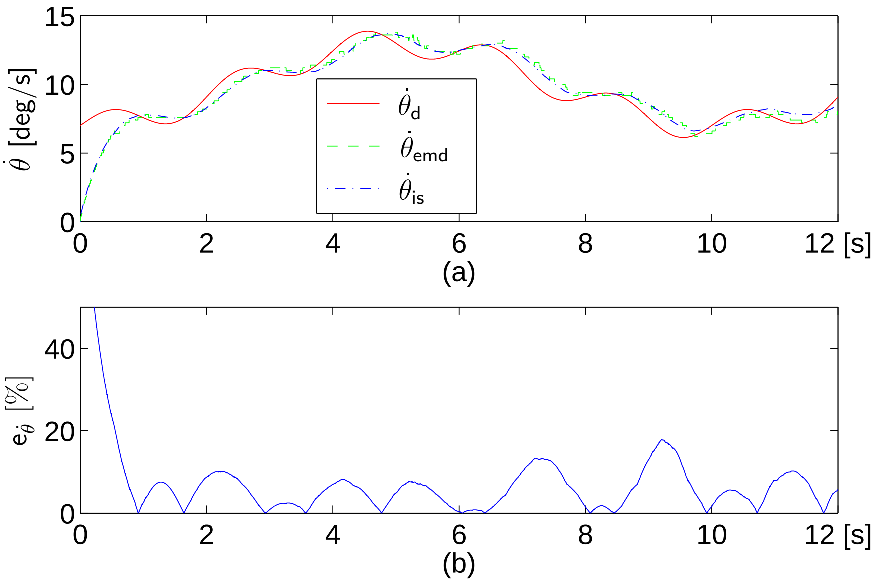


Figure 5.21: Simulation of using the mean EMD response as velocity estimator in a closed loop yaw rate control. The desired yaw rate varies as a function of time: $\dot{\theta}_\mathrm{d}(t) = 3\sin(2\pi 0.1t - \frac{\pi}{2}) + \sin(2\pi 0.5t) + 10,\ t \geq 0$. a): yaw rate evolution. Solid red-line: desired yaw rate; dashed green-line: estimated yaw rate based on EMD; dashed-dot blue-line: actual yaw rate of the manipulator; b): relative error evolution according to (5.10).

a 7-DOF robot arm [224] is utilized as a manipulator for yaw angle control. A camera-in-hand structure is adopted for image capture. A high-speed camera, Mikrotron MC1363 ($1024 \times 4\,\mathrm{pixels} @ 1000\,\mathrm{fps}$), is mounted on the end-effector. The images are sent to PC through camera link. Running the image processing algorithm with EMD takes less than 1 ms. The estimated yaw rate is sent through the network to the controller. The robot arm is controlled through the MATLAB/SIMULINK blocksets at 1000 Hz. Standalone real-time code is generated directly from the SIMULINK modules with the Real-time Workshop. The control signal for the joint is sent to the control box through a Sensoray card (Sensoray S626 I/O). The parameters of the EMD model are

$$\tau_L = 0.035\,\mathrm{s}, \quad \tau_{H'} = 0.06\,\mathrm{s}, \quad \tau_{L'} = 0.2\,\mathrm{s}, \quad \Delta\phi = 0.42\,^\circ.$$

The parameters of the PD controller are heuristically selected

$$K_p = 0.16, \quad K_d = 0.015.$$

The experiments are carried out with a normal laboratory surrounding. Similar to the simulation, two types of desired trajectories are tested: one with a constant velocity and the other one with varying velocity.

Experimental Results

Experiment I: The proposed approach for velocity estimation is evaluated firstly in a closed loop yaw angle control with a desired triangular trajectory (piecewise constant velocity), see the solid line in Fig. 5.24(a). The tracking results and the relative tracking

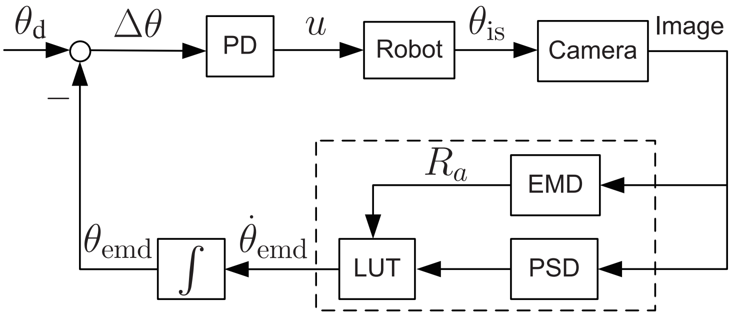


Figure 5.22: Closed loop yaw angle control system. EMD-based angular velocity estimation is denoted by the dashed rectangular. θ_d: desired yaw angle; θ_emd: estimated yaw angle; θ_is: actual yaw angle of the end-effector; $\Delta\theta$: control error; u: control signal.

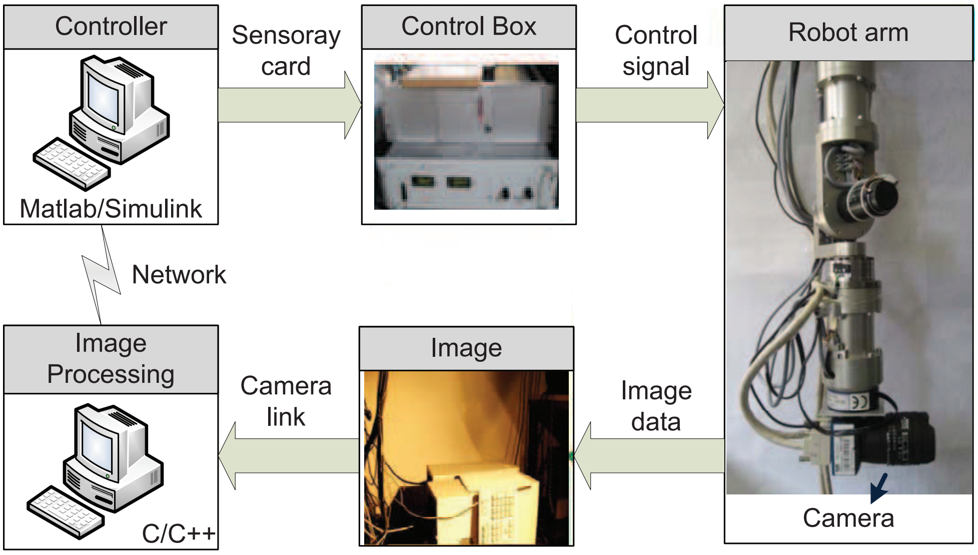


Figure 5.23: Experimental setup consisting of two standalone PCs, a robot arm, a control box and a high-speed camera mounted on the end-effector of the robot arm.

error are shown in Fig. 5.24(a) and Fig. 5.24(b), respectively. The mean relative angular error $\bar{e}_\theta$ is only 1.85%, and the maximum relative error is $\max(e_\theta) = 7.1\%$ at time instant $t = 35.6\,\mathrm{s}$. The relative angular error becomes larger when the motion direction of the desired yaw rate switches, e.g. from clockwise to anticlockwise. At these time instances, the lag due to transient EMD response and the friction on the manipulator jointly influence the yaw angle control. Nevertheless, the results above exhibit a promising start of utilizing insect-inspired EMD for accurate velocity estimation in robot control.

Experiment II: The proposed approach is evaluated in a closed loop yaw angle control with a more complex reference trajectory, which has varying acceleration and deceleration, see the solid line in Fig. 5.25 (a). Compared with the results in Fig. 5.24, the tracking error in Fig. 5.25 is slightly larger. The mean relative angular error is $\bar{e}_\theta = 1.93\%$, and the maximum relative angular error is $\max(e_\theta) = 11.7\%$. One reason for the increased relative error could be that the non-linear friction on the manipulator has a larger impact on the trajectory with varying velocity than on that with constant velocity. Moreover, the lag

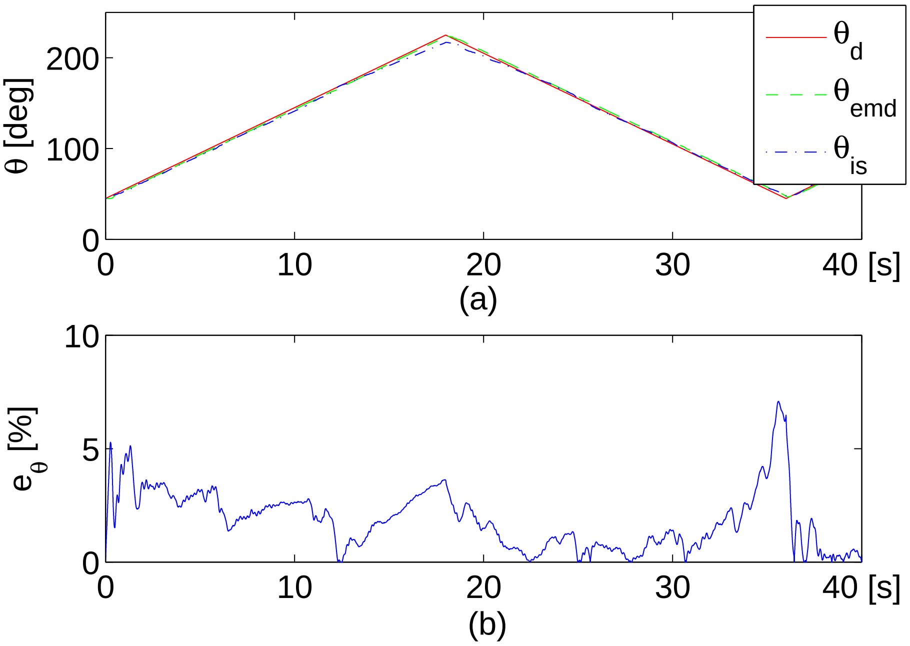


Figure 5.24: Experimental results of using the mean EMD response as velocity estimator in a closed loop yaw angle trace (with piecewise constant yaw rate) control. a): angle evolution. Solid line: desired yaw angle with piecewise constant rate; dashed line: estimated yaw angle based on EMD; dashed-dot line: actual angle of the manipulator; b): relative angle error evolution.

due to the transient EMD becomes more critical when the reference trajectory has varying velocity.

The results from these two experiments indicate that it is practical to use the EMD as a velocity estimator for robot control. The vision system based on EMD provides accurate and high-speed (1000 Hz) visual feedback. A convincing control performance is demonstrated by applying EMD in a closed loop yaw angle control. For either stepping velocity tracking or varying velocity tracking, the mean relative error is smaller than 2%.

5.3 Stable Visual Servoing with EMD

In the previous section, the raw EMD module has been enhanced for motion detection, and a novel motion estimation algorithm based on real-time LUT has been proposed for accurate motion estimation. This section is concerned with the stability problem of visual servo control system having long image processing delay in the feedback. From the evidence of the study on fly's vision system, it is found that the optomotor system of the fly does not become unstable when its overall gain gets increasingly large [225]. In this section, inspired from the fly's vision system a biologically inspired motion detector (Reichardt-model) is introduced into visual servo control to ensure the stability of a system with high feedback gains and time delays. As a consequence of the specific velocity dependence of the Reichardt-model, the stability margin of the visual servo control is increased and high overall gains, thus, better performance are achievable. In this section, visual servo control based on Reichardt-model with different sampling rates and different time delays in its

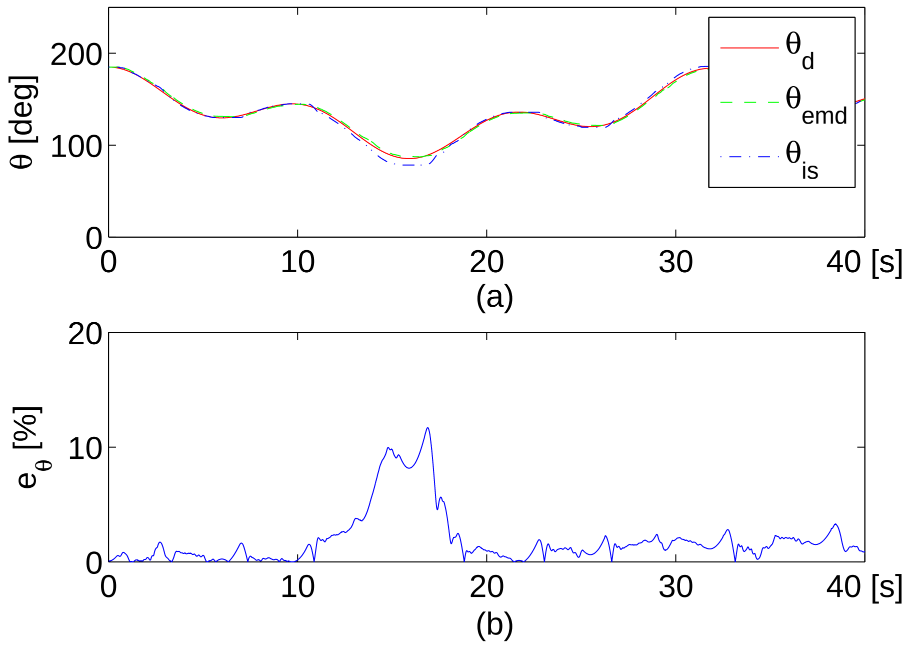


Figure 5.25: Experimental results of using the mean EMD response as velocity estimator in a closed loop yaw angle (with varying yaw rate) control. a): angle evolution. Solid line: desired yaw angle trace with varying velocity; dashed line: estimated yaw angle based on EMD; dashed-dot line: yaw angle of the manipulator; b): relative angle error evolution.

visual feedback loop is investigated. The proposed approach is evaluated both through simulations and experiments conducted on a 1-DOF linear axis with high-speed (up to 500 Hz) visual feedback.

In the rest of this section the application of Reichardt-model in inner velocity control loop of visual servo control for object tracking is firstly introduced in Section 5.3.1. Then the proposed approach is evaluated by simulations and experiments conducted on a 1-DOF linear-axis module in section 5.3.2.

5.3.1 EMD in Motion-based Visual Servoing

Since the Reichardt-model calculates the motion from image sequences, in this work the stability problem of velocity control with visual servoing (or the so called motion-based servo control) is considered.

Control Problem

Different from the control of the yaw angle in image plane in Section 5.2, a tracking system with eye-in-hand camera through motion-based visual servo control is studied in this section. Consider a serial n-link rigid robot manipulator, its dynamics can be written based on Euler-Lagrangian formulation

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} + \mathbf{g}(\mathbf{q}) = \mathbf{\Gamma},$$

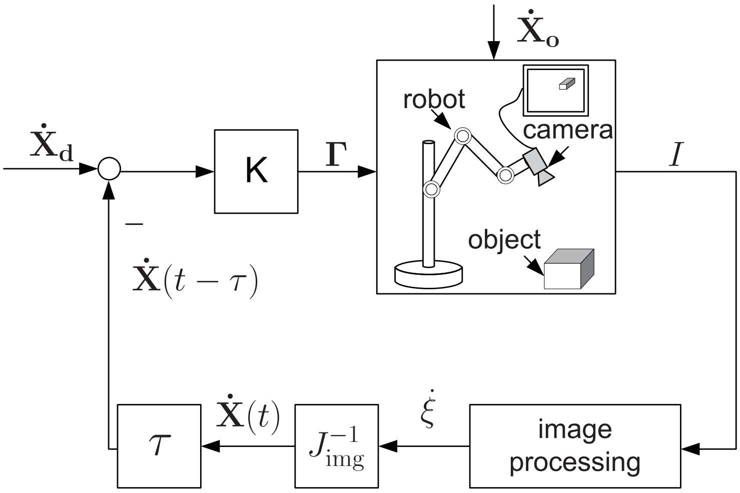


Figure 5.26: Block diagram of velocity control visual servoing for object tracking. $\mathbf{\Gamma}$: Joint torque; $\dot{\mathbf{X}}_\mathbf{o}$: motion trajectory of the target; $\dot{\mathbf{X}}$: relative motion between the camera and the target; $\dot{\mathbf{X}}_\mathbf{d}$: desired relative motion between the camera and the target; I: image sequence captured by the camera; $\dot{\xi}$: feature motion in image plane (optical flow); J_img: image Jacobian; τ: feedback delay; K: feedback gain.

where $\mathbf{q} \in \mathbb{R}^n$ is the displacement vector of joints, $\mathbf{\Gamma} \in \mathbb{R}^n$ is the torque applied to the join actuators, $M(\mathbf{q}) \in \mathbb{R}^{n\times n}$ is symmetric positive definite manipulator inertia matrix, $C(\mathbf{q}, \dot{\mathbf{q}})\dot{\mathbf{q}} \in \mathbb{R}^n$ is centripetal and Coriolis torques, and $\mathbf{g}(\mathbf{q}) \in \mathbb{R}^n$ is gravitational torques. Through direct kinematics of the robot, the pose including position and orientation of the camera frame $\mathbf{X}_\mathbf{c} \in \mathbb{R}^6$ is expressed with respect to the joint positions. For simplicity, the Cartesian space control is assumed.

The aim is to control the relative motion $\dot{\mathbf{X}}(t) \in \mathbb{R}^6$ between the camera and a moving object defined by

$$\dot{\mathbf{X}}(t) = \dot{\mathbf{X}}_\mathbf{c}(t) - \dot{\mathbf{X}}_\mathbf{o}(t), \tag{5.11}$$

where $\mathbf{X}_\mathbf{c}(t) \in \mathbb{R}^6$ and $\mathbf{X}_\mathbf{o}(t) \in \mathbb{R}^6$ denote the positions and orientations of the camera and the moving target in the robot frame, respectively. The stability of the closed loop system with large feedback gain and an amount of feedback delay is considered in system design.

Control Law

The block diagram of standard motion-based visual servo control systems is shown in Fig. 5.26. Assume a target moves in the working space with motion trajectory $\dot{\mathbf{X}}_o(t)$. Through camera perspective projection, image processing and velocity transformation, the relative motion between the object and the camera $\dot{\mathbf{X}}(t)$ is obtained with time delay τ. As presented in [226], the mapping of optic flow in the image plane (motion of feature points ξ in 2D image plane) to $\dot{\mathbf{X}}$ can be described by

$$\dot{\mathbf{X}}(t-\tau) = J_\text{img}^{-1}(\xi, z_c)\dot{\xi}(t). \tag{5.12}$$

where $\xi \in \mathbb{R}^2$ denotes the object's feature center in the image plane, $z_c < 0$ is the depth information of the object point in the camera frame, τ denotes the feedback delay, and

J_{img} is the so-called image Jacobian defined by

$$J_{\text{img}}(\xi, z_c) = \begin{bmatrix} \frac{\alpha\lambda}{z_c} & 0 & \frac{x}{z_c} & -\frac{xy}{\alpha\lambda} & \frac{\alpha^2\lambda^2+x^2}{\alpha\lambda} & y \\ 0 & \frac{\alpha\lambda}{z_c} & -\frac{y}{z_c} & -\frac{\alpha^2\lambda^2+y^2}{\alpha\lambda} & \frac{xy}{\alpha\lambda} & -x \end{bmatrix}. \tag{5.13}$$

Here, α and λ are intrinsic parameters of the camera. The depth information z_c can be obtained either from multi cameras, knowledge of the geometric relationship or additional external sensors. It should be mentioned, additional rotation matrix for the displacement between camera frame and robot frame, and transformation matrix relating camera angular velocity to the time derivative of the Euler angles (or other angles representations) are omitted in (5.12) for simplicity.

After obtaining the current relative motion between the camera and the target, the following control law is then applied

$$\mathbf{u}(t) = -K(\dot{\mathbf{X}}(t-\tau) - \dot{\mathbf{X}}_{\mathbf{d}}(t)), \tag{5.14}$$

where $\dot{\mathbf{X}}_{\mathbf{d}}(t)$ is the reference of the desired motion, $K \in \mathbb{R}^6$ is a symmetric positive definite proportional matrix chosen by the designer.

Notice a time delay τ due to data transmission and image processing exists in the feedback loop. The system control performance and the system stability are related with the feedback gain K and the feedback time delay τ.

Velocity Estimation based on Reichardt-model

There are different methods for calculating feature velocity (optical flow) from image sequences, e.g. SSD algorithm [227], Lucas&Kanade method and Horn&Schunck method [228] and so on. However, by introducing the optical flow calculation into the loop, time delay (usually tens/hundreds of millisecond) due to camera exposure time, time for image transferring and image processing, and time for data transmission (if a distributed system is selected) jointly limits the control gains and, thus, the performances. When the control gain K increases in presence of latency in the feedback loop, the system given by Fig. 5.26 will become unstable. In order to overcome this drawback, the biologically inspired motion detector Reichardt-model is applied to stabilize the system.

An example of introducing the Reichardt-model into velocity control loop is shown in Fig. 5.27. Different from conventional image processing algorithms, optical flow in the system shown in Fig. 5.27 is estimated based on Reichardt-Model. The image sequence I is captured by the camera. The Reichardt-model is then applied to each pixel to calculate the response. Then a LUT is applied for velocity estimation. Since the focus here is on the system stability, a static LUT is applied in the feedback loop to simplify the system design. After obtaining the optical flow $\dot{\xi}$ in the image plane, the relative motion of the camera $\dot{\mathbf{X}}$ is calculated by (5.12) with a latency of τ. Subsequently, the control signal for the robot is determined according to (5.14).

5.3.2 Performance Evaluation

Simulation of velocity control

In this part, the standard velocity control approach in Fig. 5.26 and the Reichardt-based velocity control approach in Fig. 5.27 are simulated. In the simulation, a 1-DOF linear-axis

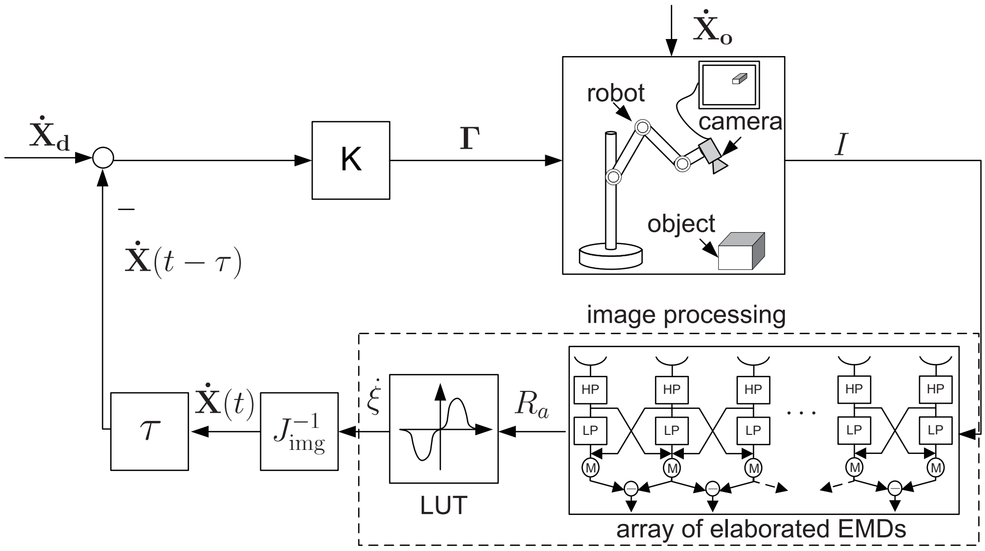


Figure 5.27: Block diagram of visual servoing with the Reichardt-model in the feedback loop. R_a: response of elaborated EMD array; LUT: lookup table of velocity-response dependency.

determined from experiments

$$0.4\,\ddot{q} + 0.3\,\dot{q} = \tau,$$

is modeled as the manipulator. The target is set to have a moving velocity $\dot{X}_o = 0.1\,\mathrm{m/s}$. Assume the desired relative motion $\dot{X}_d$ of the camera with respect to the object is zero, $\dot{X}_d = 0$.

The time delay module with $\tau = 20\,\mathrm{ms}$ is added in the feedback loop, which simulates the image processing and transmission delays. Two different feedback gain sets are tested, namely, small feedback gains $K = 1, 10$, and large feedback gains $K = 47$.

The simulation results are shown in Fig. 5.28. Fig. 5.28(a)(b) show the velocity trajectories of the end-effector when the feedback gains are small ($K = 1$, or 10) in both approaches. The curves 'S_1' and 'S_10' are the velocity evolution trajectories of the standard approach when $K = 1$, 10, while 'R_1' and 'R_10' are the velocity trajectories of the Reichardt-based approach. In this case, both systems settle down in steady state. However, when the feedback gain increases, e.g. $K = 47$, the system without Reichardt-model becomes unstable, see curve 'S_47' in Fig. 5.28(c), while the system with Reichardt-model is still under control with certain oscillation remaining, see curve 'R_47' in Fig. 5.28(c).

The simulation results show that when time delay is present and the overall feedback gain increases, the velocity control loop with Reichardt-model remains stable due to the intrinsic properties of the Reichardt-model, while the standard approach becomes unstable. The reason is that, utilizing the Reichardt-model for the estimation of optical flow plays an additional role: dynamical tuning of the feedback gain. In other words, the output of Reichardt-model becomes small when the input signal has a large velocity. It has a similar effect as decreasing the feedback gain which stabilizes the system.

Experiments

In this part, the velocity control models simulated above are tested by experiments conducted on two 1-DOF linear-axis. The target is mounted on one linear-axis module, while

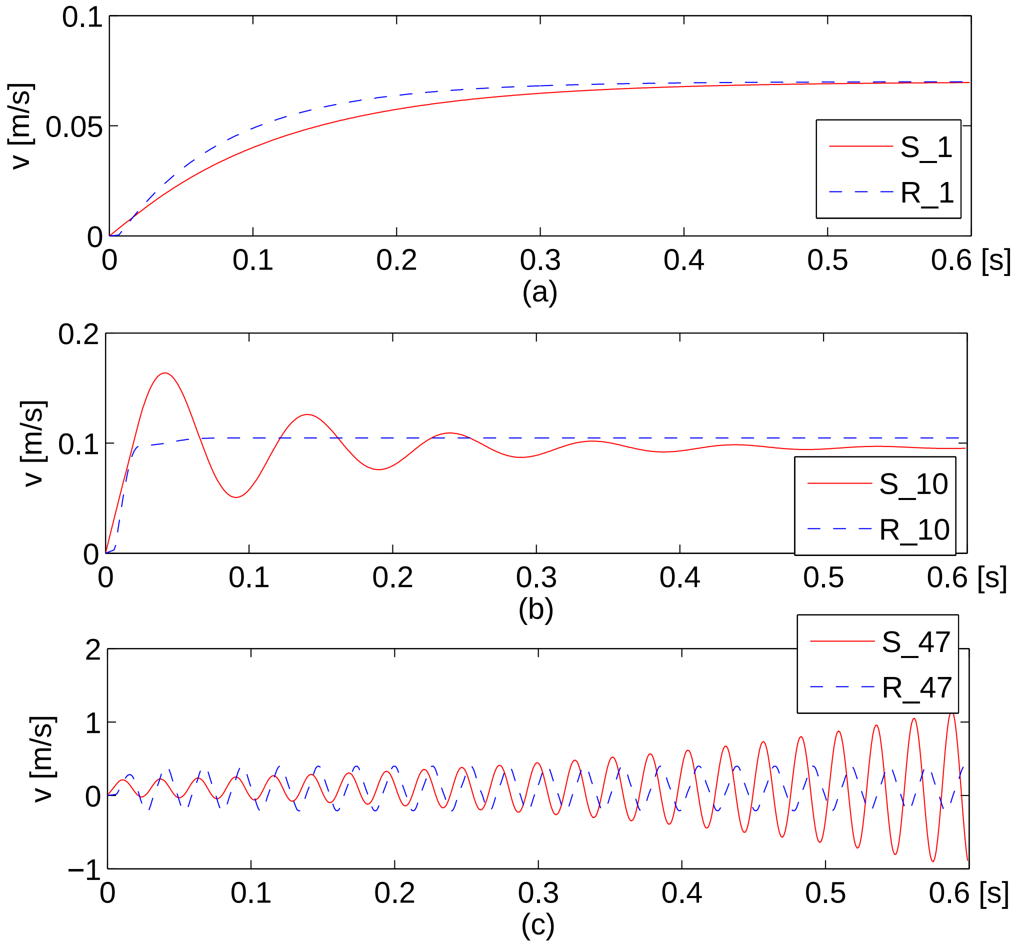


Figure 5.28: Velocity trajectories of the camera in the robot frame with standard/Reichardt-based approach. (a),(b): results of systems with small feedback gains $K = 1, 10$; (c): results of systems with large feedback gain $K = 47$. Solid lines: results of standard approach; dashed lines: results of the Reichardt-based approach.

.

the camera is mounted on the other one, see Fig. 5.29(a). Two sets of experiment are designed, and the experimental setup and results are described in the following part.

Experimental Setup A linear motor module shown in Fig. 5.29(b) is selected as the basic module. The system control is implemented using Simulink/Realtime workshop, which runs on a PC (i686, AMD 64 Processor 3000+) with fixed sampling period of 1ms. A high-speed CMOS imaging camera Mikrotron MC1319 is mounted on the forcer. It provides high speed imaging with up to 500 full frames per second. The camera runs at a frame rate of 30 fps and has a resolution of 640 × 480 pixels. The other intrinsic parameters of the cameras are as follows: focal length $\lambda = 17$ mm, scaling factor $\alpha = 88,888$ pixels/m, see also Tab. 5.3. The captured image is transferred to a host PC (x86_64, AMD 64x2 dual core Processor 5200+) through Camera Link. The velocity of the target $\dot{X}_o$ is set to be 0.1 m/s and the desired relative motion is zero ($\dot{X}_d = 0$).

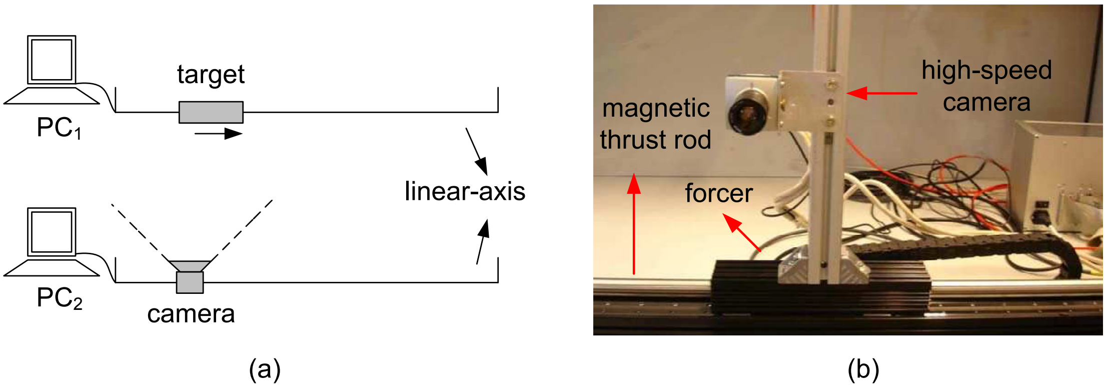


Figure 5.29: Experimental Setup on linear-axis modules. (a): structure of the motion-based control system on two 1-DOF linear-axis modules; (b): experimental setup with high-speed camera mounted on the linear-axis.

Table 5.3: Parameters of the camera and PCs.

Camera MC1319	frame rate	30 fps
	resolution	640×480 pixels
	focal length λ	17 mm
	scaling factor α	88,888 pixels/m
PC for imaging	x86_64, AMD 64x2 dual core Processor 5200+	
PC for control	i686, AMD 64 Processor 3000+	

Two sets of experiment are carried out. Firstly, the standard velocity control model and the Reichardt-based model are tested with different feedback gains (with small constant time delay in the feedback). Then, different length of time delay is added to the feedback loop (with small constant feedback gain). The control performances of the two approaches in these experiments are presented and compared.

Experiment Results I: Different Feedback Gains Similar to the simulation, the standard approach and the Reichardt-based approach are tested with different feedback gains. In this experiment, the time delay in the feedback loop is set to be $\tau = 20\,\text{ms}$. A small feedback gain $K = 2$ and a relatively large feedback gain $K = 7$ are selected and tested in the experiments.

The experimental results are presented in Fig. 5.30. The velocity evolutions of the two approaches are shown in Fig. 5.30(a)(b), while the position evolutions are shown in Fig. 5.30(c)(d). The red-line denotes the results of the standard approach, while the dashed blue-line denotes the results of the Reichardt-based approach. When the feedback gain is small (e.g. K= 2) both models are stable, see the curves 'S_2' and 'R_2' in Fig. 5.30(a)-(d). However, the standard motion-based visual servoing system becomes unstable when the feedback gain increases to $K = 7$, see the curves 'S_7' in Fig. 5.30(a)(c). The velocity of the end-effector becomes larger and larger instead of settling down. For the system with the Reichardt-detector in its feedback, it is stable, as shown in Fig. 5.30(b)(d). Only small oscillations are present.

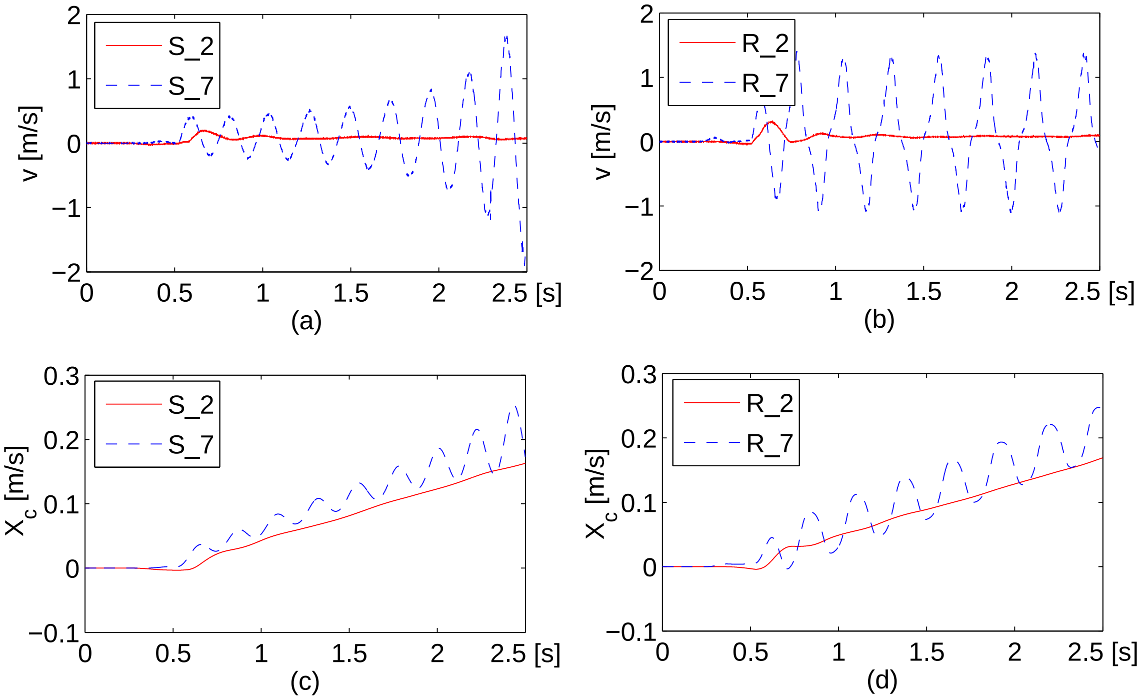


Figure 5.30: Velocity and position evolutions of the camera in standard/Reichardt-based velocity control system with visual feedback, tested with different feedback gains $K = 2$, 7. (a),(b): velocity evolutions; (c),(d): position evolutions. Solid lines: results of standard approach; dashed lines: results of Reichardt-based approach. $\tau = 20\,\text{ms}$.

It is concluded from the experimental results that, the Reichardt-model prevents the system from getting unstable when the overall feedback gain increases and small time delay is present in the feedback loop.

Experiment results II: Different Time Delays In this experiment, the feedback gain is set to be constant ($K = 2$) and different length of artificial time delays is added into the feedback loop: $\tau = 50\,\text{ms}, 160\,\text{ms}$. The experimental results are shown in Fig. 5.31. Similar results are observed. When time delay is small, e.g. $\tau = 50\,\text{ms}$, both models are stable, see curves 'S_50' and 'R_50' in Fig. 5.31(a)-(d). When the time delay in the feedback is increased to 160 ms, the velocity of the camera in the standard approach increases rapidly and the system tends to become unstable, see curve 'S_160' in Fig. 5.31(a)(c). In contrast, the system is still stable when the Reichardt-model is applied in the closed loop, see curve 'R_160' in Fig. 5.31(b)(d).

It is observed that remaining oscillations are present with the Reichardt-based closed loop control in both experiments which e.g. could be limit cycles of the quantized and/or nonlinear system. This effect is subject to further investigations.

In this section, the biologically inspired motion detector Reichardt-model is introduced into visual servo control. As a consequence of the specific response-velocity dependence of the Reichardt-model, the control system with the Reichardt-model in the feedback loop is demonstrated to have increased stability margin and allow higher overall gains. A

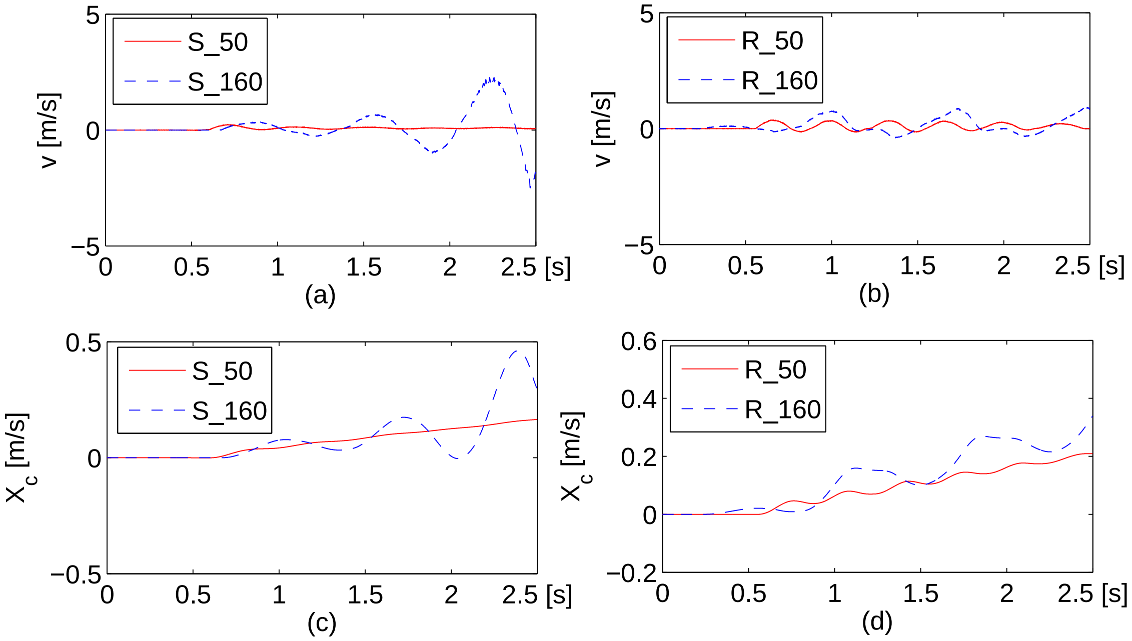


Figure 5.31: Velocity and position evolutions of the camera in standard/Reichardt-based velocity control system with visual feedback, tested with different feedback delays $\tau = 50$ ms, 160 ms. (a)(b): velocity evolutions; (c)(d): position evolutions. Solid lines: results of standard approach; dashed lines: results of Reichardt-based approach. $K = 2$.

Reichardt-based control scheme is compared with the conventional motion-based visual servo approach. Both approaches are simulated and tested in real-time experiments on 1-DOF linear motor modules. The simulation and experimental results demonstrated that the Reichardt-based approach can prevent the system from getting unstable when feedback gain increases and large time delay is present.

5.4 Discussion

In order to overcome the low sampling rate problem of visual feedback, the insect's vision system, which is simple and yet, presents promising perception capability, innovates the design of biologically inspired approaches for high-speed vision in robot control. The well-known Reichardt-model has been widely accepted as the motion sensitive detector in fly's vision system. However, this model has been primarily utilized as a qualitative motion detector rather than a quantitative motion detector due to its low accuracy in motion estimation. Besides, fly's vision system with Reichardt-model does not get unstable even with high feedback gains because of its bell-shaped response-velocity relationship. It inspires the design of stable visual servo control system by introducing the Reichardt-model into the closed loop system.

In this chapter, a novel motion estimation approach has been proposed based on the insect-inspired EMD. It focuses on the response of EMD to real-time image sequence in an unconstructed environment. An elaborated model and several steps has been proposed to optimize the EMD as a quantitative motion detector. A real-time response-velocity

LUT is established based on the analysis of the power spectrum of the input image. The proposed approach is evaluated both in simulation and experiments of closed loop control. The results demonstrated that the velocity estimated by the proposed approach is accurate and thus can be applied in real-time robot control. The proposed approach for velocity estimation has the advantage of low computation cost, and shows a significant improvement of the robustness in velocity estimation.

In order to stabilize a visual servo control system with high feedback gains and delays, a novel Reichardt-model based approach has been proposed in this work. As a consequence of the specific velocity dependence of the Reichardt-model, the control system with the Reichardt-model in the feedback loop has been demonstrated to have increased stability margin and allow higher overall gains, resulting in better performance. The proposed approach and the conventional approach have been simulated and evaluated in real-time experiments on 1-DOF linear motor modules. It is demonstrated that the proposed approach prevents the system from getting unstable when the feedback gain increases.

The benefits of the proposed approaches introduced in this chapter are summarized below:

- The performance of motion estimation based on Reichardt-model is improved. It facilitates the implementations of insect-inspired robotics.
- Increased stability margin is achieved with the involvement of the Reichardt-model in the closed loop.

In this chapter, the biologically inspired motion detector has been combined with the conventional visual servo control and applied for robot control for the first time. Key issues of insect-inspired motion perception and closed loop control are covered. The results provide a fundamental understanding of high-speed insect-inspired vision and contribute to the further development of high-speed vision-based control in the robotics domain. In system design, only the ascending part of the response-velocity curve is used in order to eliminate the ambiguity in velocity estimation. How to enlarge the usage scope by dynamically tuning the parameters of the filters online is still an open problem. Furthermore, EMD-based velocity estimation for more than one dimension combined with receptive fields is not addressed in this thesis, which is subject to future work.

6 Conclusions and Future Directions

6.1 Concluding Remarks

Visual feedback provides the necessary perception information for robot control. Therefore, it is expected to be reliable, fast and robust. In most vision-based control systems, the control performance is often affected by the low-sampling-rate visual feedback due to limited computation capacity and real-time constrains.

This thesis focuses on the methodical investigation of high-speed vision for vision-based robot control. In order to increase the sampling rate of the visual feedback, two novel approaches, the networked-based approach and the insect-inspired approach, are proposed in this thesis. In the first approach, the existing networked computation resources are considered in building a could computing platform for parallel image processing. In the second approach, high-speed vision system is established based on an insect-inspired motion detector with a simple structure. The main contribution of this article are i) a could computing platform based on a real-time image transmission protocol, ii) a robust image processing algorithm, iii) a control law stabilizing the networked visual servo control system with time-varying feedback delays, iv) a sending rate scheduling strategy aiming at reducing the communication network load, and v) an insect-inspired visual servo control system with the capability of accurate velocity estimation and increased stability margin. The main approaches along with the major results are highlighted in the following.

For highly dynamic vision-based motion control, high-speed visual feedback is essential to the performance of the system. For accelerating visual feedback, the computation resources connected over a communication network is less considered in known works. Distributing images to processing nodes over a communication network results in networked visual servo control. Transmitting data over the network and distributed image processing are known in literature. However, high-speed vision over the communication network results in a large volume image data transmission. Distributed image processing for real-time closed loop control is not covered in the literature. Moreover, the robustness of the image processing algorithm is another important issue for high-performance control. The feature matching algorithm, which is a critical step of image processing, is yet not optimal in particular for the tracking problem with long distance object. In Chapter 3, a real-time transport protocol ncRTP is developed for large volume image data transmission over a communication network. Based on this protocol a cloud computing platform is established, which consists of a streaming server, several processing nodes, and a controller. In this context, images are distributed by the streaming server to the processing nodes available over a communication network, and the processing results are sent to the controller generating configuration signals for manipulators. The newly developed platform facilities the implementation of a networked vision system that is capable of providing high-sampling-rate visual feedback for. Additionally, a novel feature matching algorithm is proposed to improve the robustness of image processing. In order to increase the matching rate and to reduce the matching error, the sampling distribution in the scale space for feature extraction is analyzed. The features are firstly extracted in the scale space with coarse evenly-spaced sampling, and after that the average scale of the object is roughly determined online. Then, a binomial distributed sampling around the average scale is proposed for the second step of feature detection. The determined average scale is also utilized in feature matching. The results show an improved matching performance in particular for the tracking of objects with long distance.

In spite of significant improvements in image processing capability, networked visual servo control challenges system design. In contrast to conventional visual servoing, in the networked visual servo control system the image transmission over the communication network induces time-varying time delay in addition to computation time delay. The feedback time delay leads typically to a control performance degradation, and may even cause unstable closed loop behavior. In addition, transmitting image data over the communication network, which has constrained network resources, increases the probability of network congestion and packet loss fatal to the control system. In Chapter 4, the networked visual servo control system with aperiodic sampling intervals and time-varying transmission and image processing time delays is reformulated into a continuous-time system with random time-varying time delay by an input-delay approach. The occurrence probabilities of the time-varying time delays are described by a set of indicator functions having independent and identical distribution (i.i.d). The resulting system is a continuous-time system with a controller switching the feedback gains depending on the time-varying feedback time delay. Associated stability constrains and controller design are derived based on the probabilistic distributions of the sampling intervals and delays. Since image transmission over the communication network results in a high network load, and may even lead to a complete communication blackout, the trade-off between the control performance and the network usage is concerned. The focus here is to design a sending rate scheduling strategy which preserves a guaranteed control performance and meanwhile efficiently utilizes the network resources. Therefore, a stochastic cost function with respect to the probability distributions of sending intervals, which considers jointly the control performance and the network cost, is proposed. By optimizing the cost function the probability distributions of different sending intervals are determined, and then utilized in sending rate scheduling. For the tracking problem studied in this thesis, the tracking error is additionally considered for the adjustment of the sending rate. Benefits in terms of guaranteed control performance and reduced network load are demonstrated in the experiments conducted on 7-DOF robotarm.

While Chapter 3 and 4 focus on the network-based approach, further investigation of high-speed vision with the inspiration from neurobiology is also of great interest. In Chapter 5, the well-known Reichardt-model, which is widely accepted as motion sensitive detector in fly's vision system, has been introduced into the feedback loop of visual servo control system. Since the raw Reichardt model has a poor performance in quantitative velocity estimation, an elaborated Reichardt model is proposed in this chapter by investigating its responses to single sinusoid, multi-sinusoids, and real-time image sequence. The power spectrum of the input signal is extracted, and its influence on the response of Reichardt model is analyzed. Based on this information, a novel LUT-based approach is proposed for velocity estimation. In order to improve the performance, the fundamental effects of the system parameters such as the field of view of the camera, the image resolution, and the filter constants, are analytically investigated through simulations. The proposed approach has the benefits of simple structure and thus enables high-speed visual feedback. Furthermore, the intrinsic property of the Reichardt model, namely the bell-shaped velocity-response relationship, is utilized to stabilize the closed loop system. The experimental results shown that the proposed system has an increased stability margin and allows higher feedback gains, thus, resulting in a better control performance.

In summary, the benefits of the networked visual servo control proposed in this thesis are the enhancement of the computation capability, the efficiency and the flexibility of system reconstruction, the improvement of image processing in terms of the robustness to distance, the guaranteed control performance, and the efficient usage of network resources. The insect-inspired visual servo approach has combined the biologically inspired method with the conventional approach for the first time. It will facilitate the implementations of insect-inspired robot control with simple structure and promising control performance. The contributions of this thesis advance the state of the art in the development of vision-based robot control and provide valuable insights for future research.

6.2 Outlook

High-speed vision becomes increasingly important for applications in robotics domain, in particular for highly dynamic vision-based motion control. Benefiting from the advances in communication and information technology, it is feasible to use the existing computational resources connected over a communication network for cloud data processing. Due to its reconfigurability and versatility, the networked high-speed vision system is considered as an alternative to the traditional vision system. Furthermore, the advances in neurobiology boost the development of insect-inspired vision systems, which outperform the conventional vision systems in terms of simplicity and low computation cost. Although the fundamental research on system design, control and cost has been investigated in this thesis, networked visual servo control and insect-inspired closed loop control are still relatively young research fields. There are a number of open questions and future directions emerging from this thesis, some of which are:

- *High-speed vision for multi visual sensor system* – Compared with the one-sensor system, multi-sensor system enables visual perception with a larger field of view and better understanding of the dynamical surroundings. The investigation of multi-sensor system has been focused in some works, which concern mainly data fusion and processing algorithms. Running the multi-sensor system in an efficient and high-speed manner has not yet been considered. In order to optimize the perception performance, coordination and cooperative computation of the resources from the multi-sensor over the network are important issues in networked high-speed vision system.

- *Nonlinear control design for networked visual servoing* – The nonlinear property of visual servo control system due to camera projection and manipulator dynamics has been ignored to a large extent in most existing works. The nonlinear system is often linearized at the equilibrium states, which leads to a limited working space. An unified approach combining more general nonlinear control and communication strategies is an important feature work in networked visual servo control.

- *Full degrees of freedom motion estimation for insect-inspired robotics* – The insect-inspired vision system presents novel merits in its simple structure and accordingly low computation cost. In order to expand the application of insect-inspired vision, high-speed full degrees of freedom motion estimation with high accuracy is a dispensable requirement for promoting deep understanding in both traditional engineering and neurobiology approaches.

The future research on high-speed vision requires multi-discipline exploration incorporating communication, information, control and neurobiology. Integrating the various aspects of these research fields cooperatively will lead to superior visual servo control with the advantages of flexibility, maneuverability, efficiency and capacity, and it will highly advance the state-of-the-art with high impact on future robotic technologies and applications.

A Dual Arm Manipulator

In this thesis a dual arm manipulator was applied in the experiments to demonstrated the proposed networked visual servo control system and insect-inspired closed-loop system. The forward kinematics and the hardware of the manipulator are introduced here.

The manipulator has two identical 7-DOF human-scaled arms, as shown in Fig. 4.12. Each arm has two spherical joints with three DOFs at the shoulder and the wrist. At the elbow, there is one rotational joint. The Devavit-Hartenberg parameters for the arm are listed in Tab. A.1, and the assignment of frames is shown in Fig. A.1.

The position and orientation of the end-effector in the base frame is describe by a homogeneous transformation matrix

$$^{7}T_0 = \begin{bmatrix} R & \mathbf{p} \\ 0 & 1 \end{bmatrix},$$

where $\mathbf{p} = [p_x \, p_y \, p_z]^T$ represents the position vector and the matrix $R \in \mathbb{R}^{3\times 3}$ denotes the orientation matrix. The velocity of the end-effector is described by

$$\begin{bmatrix} \dot{\mathbf{p}} \\ \boldsymbol{\omega} \end{bmatrix} = J(q)\dot{\mathbf{q}},$$

where $\dot{\mathbf{p}}$ and $\boldsymbol{\omega}$ denote the linear velocity and the angular velocity, respectively, and $\mathbf{q}$ is the vector of joint displacement. The matrix J is called the geometrical matrix, which is computed by

$$J = [J_1 \, J_2 \, \dots J_n],$$
$$J_i = \begin{bmatrix} \mathbf{z_{i-1}} \times (\mathbf{o_n} - \mathbf{o_{n-1}}) \\ \mathbf{z_{i-1}} \end{bmatrix},$$

where $\mathbf{z_i} = [0 \ 0 \ 1]^T$ denotes the rotation axis of the i-th link, and $\mathbf{o_i}$ is the vector from the origin of the based frame to the origin of the i-th frame.

The damped least squares (DLS) method known as the Levenberg-Marquardt [229] is utilized to determine the inverse Jacobian. In oder to overcome the problems of singularities, discontinuity, and the excessive velocities, the cost function

$$g(\dot{q}, \dot{x}) = ||\dot{x} - J\dot{q}|| + \lambda^2 ||\dot{q}^2||$$

is defined, where λ is a constant and $\dot{\mathbf{x}} = [\dot{\mathbf{p}} \ \boldsymbol{\omega}]^T$. In this cost function, the tracking error in the Cartesian space is weighted against the norm of joint velocity though the damping factor λ. The damped least squares solution is equal to

$$\dot{\mathbf{q}} = J^T(JJ^T + \lambda^2 I)^{-1}\dot{\mathbf{x}} = J^*\dot{\mathbf{x}},$$

The singular value decomposition (SVD) of the solution is

$$J^* = \sum_{i=1}^{R} \frac{\sigma_i}{\sigma_i^2 + \lambda^2} \mathbf{v_i}\mathbf{u_i}^T,$$

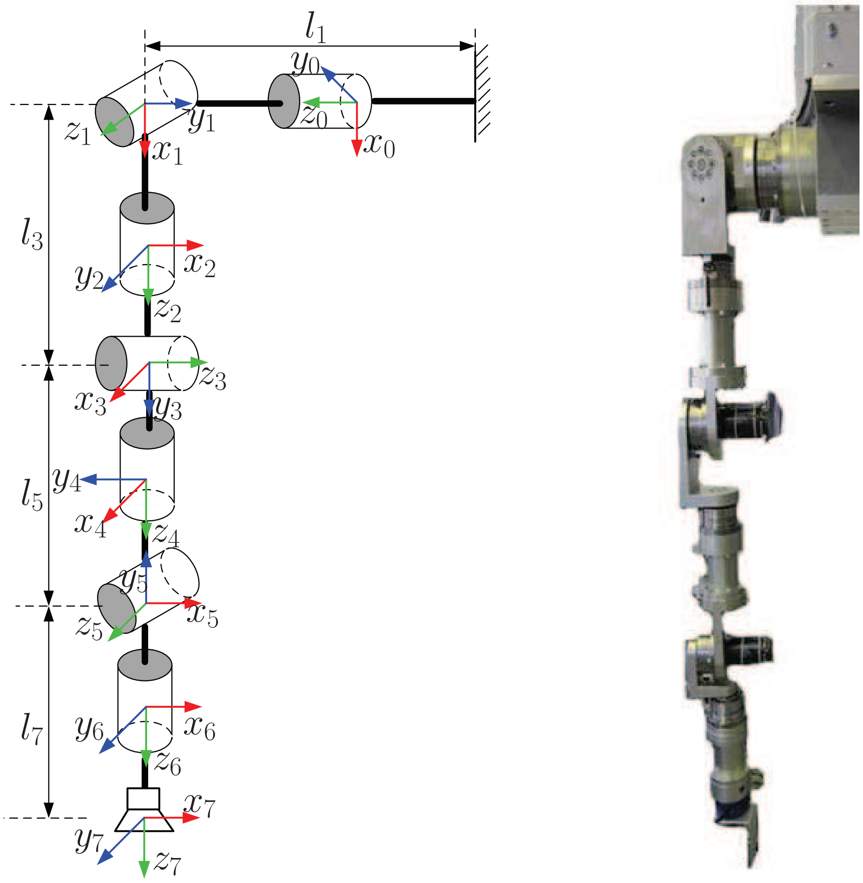


Figure A.1: Coordinate system of the 7-DOF robotarm (left) and photography of the physical arm construction (right).

where $\mathbf{u_i}$ and $\mathbf{v_i}$ denote the i-th columns of the orthogonal U, and the diagonal V matrices, respectively, and $\sigma = d_{i,i}$ denotes the diagonal entries in the matrix D. The constant velocity weighting factor λ guarantees bounded velocities and smooth transitions through singular configurations, however, with the cost of end-effector errors even when the manipulator is far away from singular configurations.

A.1 Hardware Components

The arm reach is 0.86 m, weight in total approximately 13.5 kg. The motors are Maxon motor type RE40, gears Harmonic Drive, see Tab. A.1. The motor are controlled through the I/O board Sensoray 626. The position of the joint is measured by an optic pulse incremental encoder on the motor shaft, and then processed by a quadrature encoder on the I/O board. The control loops are built with MATLAB/SIMULINK blocksets. The standard alone real-time code for RT Linux is automatically generated from the SIMULINK model.

For tracking system a high-speed camera was mounted on the end-effector, see also Fig. A.1. It runs at high frame rate ($\geq 400\,\text{Hz}$) simulating event-triggered image capturing. The camera EoSens MC1363 from Mikrotron shown in Fig. A.2 was adopted in the

Table A.1: Hardware details of the 7-DOF robotarm [192]

Joint	Link length	Link twist	Joint limits	Gear Type	Gear Reduction
i	l_i [m]	α_i [rad]	[rad]		
1	0.266	$-\pi/2$	-	HFUC-25-160	160
2	0	$\pi/2$	[0.886, 5.397]	HFUC-25-160	160
3	0.312	$-\pi/2$	-	HFUC-25-160	160
4	0	$\pi/2$	[-2.377, 2.377]	HFUC-25-160	160
5	0.312	$-\pi/2$	-	HFUC-17-100	100
6	0	$\pi/2$	[-0.174, 0.174]	HFUC-17-100	100
7	0.244	0	-	HFUC-17-100	100

Figure A.2: The MC1363 high-speed camera.

Specification	Description
Data width	8/10
Col-or/Mono	Color
Max. frame rate @ 1280×1024	500 fps
Image pre-processing	+
Interfaces	Camera Link
Camera size	63×63×47 mm

Table A.2: Key parameters of the camera.

experiments. The key parameters of the camera are listed in the Tab. A.2. More information can be found in [230]. In order to determine the intrinsic parameters (e.g. focal length, principal point, and distortion coefficients) of the camera, it is calibrated off-line before carrying out the experiments.

A.2 Computed Torque Feedforward Control

In this thesis, the 7-DOF robotarm is linearized based on the computed torqued feedback approach, which is widely used to compensate the nonlinear dynamics of the manipulator. Thus, motion of each joint can be controlled individually using other linear control strategies.

Consider a serial $n-$link robot dynamics modeled as a rigid body dynamics system

$$M(\mathbf{q})\ddot{\mathbf{q}} + C(\mathbf{q},\dot{\mathbf{q}})\dot{\mathbf{q}} + g(\mathbf{q}) = \mathbf{\Gamma}, \tag{A.1}$$

where $\mathbf{q}$, $\dot{\mathbf{q}}$, $\ddot{\mathbf{q}}$ are joint displacements, velocities and accelerations of the robot, $M(\mathbf{q})$ is the inertial matrix of the robot, $C(\mathbf{q},\dot{\mathbf{q}})$ the centripetal and Coriolis matrix, and $\mathbf{\Gamma}$ the inputs such as joint toques to the robot system. The input signal $\mathbf{\Gamma}$ is usually generated from a controller which is designed to drive the robot to follow desired trajectories.

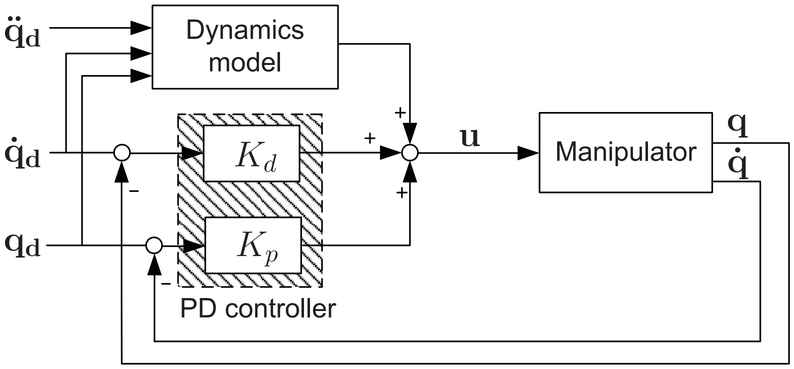


Figure A.3: Computed torque feedforward approach. q_d, $\dot{q}_d$, $\ddot{q}_d$: desired joint angle, velocity and acceleration.q, $\dot{q}$, $\ddot{q}$: measured data. K_p, K_d: control parameters. u: control signal.

The intension of feedforward approach is eliminating the nonlinearity in the dynamics of the robot as shown in Fig. A.3. In order to execute the tracking task a simple independent joint PD controller is applied to the linearized system

$$\mathbf{u}^* = K_p\mathbf{e} + K_d\dot{\mathbf{e}}, \tag{A.2}$$

where $\mathbf{e} = \mathbf{q_d} - \mathbf{q}$, $\mathbf{q_d}$ and $\dot{\mathbf{q}}_\mathbf{d}$ are desired joint positions and velocities, $\mathbf{u}^*$ is the vector of control input, K_p and K_d denote $n \times n$ diagonal matrices of velocity and position gains. Combining (A.1) and (A.2) yields the command for the robot

$$\begin{aligned}\mathbf{u} &= M(\mathbf{q}_d)\ddot{\mathbf{q}}_\mathbf{d} + C(\mathbf{q}_d, \dot{\mathbf{q}}_\mathbf{d})\dot{\mathbf{q}}_\mathbf{d} + g(\mathbf{q_d}) + K_p\mathbf{e} + K_d\dot{\mathbf{e}} \\ &= f(\mathbf{q_d}, \dot{\mathbf{q}}_\mathbf{d}, \ddot{\mathbf{q}}_\mathbf{d}) + K_p\mathbf{e} + K_d\dot{\mathbf{e}},\end{aligned}$$

where $f(\cdot)$ denotes the feedforward term, referred as a set of nominal torques allowing linearized dynamics around the operating positions $\mathbf{q_d}$, $\dot{\mathbf{q}}_\mathbf{d}$ and $\ddot{\mathbf{q}}_\mathbf{d}$. For saving the computation time, the feedforward term with only parameters of desired trajectories involved can be calculated offline. If the dynamics of the robot is available and exact, each link of the robot is decoupled and can be controlled individually. By chosen appropriate gain matrices K_p, K_d the tracking error is converges to zero.

It has to be mentioned that the computed torque control approach requires exact dynamical knowledges of robot manipulators, which is difficult to obtain in practical situations, and may result in performance devaluation.

B Design Tools

As introduced in Chapter 4, the derived controller design algorithms are represented in terms of LMIs. In order to solve the LMI problems, a numerical toolbox `Yalmip Toolbox` [187] in Matlab is introduced in this section. In addition, a network emulator is described which is involved in the experiments of networked high-speed visual servo control. Some useful links related to LMI optimization and the network emulator NetEm are given below:

- Yalmip Toolbox, *http://users.isy.liu.se/johanl/yalmip*,
- PENBMI, *http://www.penopt.com.*
- Netem, *http://linux-net.osdl.org/index.php/Netem*

B.1 Optimization Tool

Yalmip a toolbox for modeling and optimization in MATLAB. It was initially indented for probably semidefinite programming (SDP) and linear matrix inequalities (LMI), and now also supports linear programming (LP), quadratic programming, quadratic programming (QP), semidefinite programs with bilinear matrix inequalities (BMI) and so on. In the following, two numerical examples of using Yalmip and PENBMI for controller design and cost function optimization are given.

Example 1 Consider a NVSC system with the dynamics described by

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ -1 & -15 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1.5 \\ 1 \end{bmatrix} \mathbf{u}(t). \tag{B.1}$$

The i.i.d. transmission delay, computation delay and sampling intervals are bounded by $\tau_k^x \in [5, 10]\,\mathrm{ms}$, $\tau_k^c \in [10, 45]\,\mathrm{ms}$, and $d_k \in [5, 20]\,\mathrm{ms}$, respectively. The values of the parameters used in the simulations are listed in Tab. B.1. The overall feedback delay

Table B.1: The parameters for simulation of NVSC system with switching control.

Boundaries of the delay intervals s_i	Occurrence probabilities p_i	Feedback gains K_i
$s_1 = 35\,\mathrm{ms}$	$p_1 = 65\%$	$K_1 = [-23.36, -16.31]$
$s_2 = 50\,\mathrm{ms}$	$p_2 = 25\%$	$K_2 = [-12.28, -8.71]$
$s_3 = 75\,\mathrm{ms}$	$p_3 = 10\%$	$K_3 = [-5.36, -3.82]$

is categorized by s_i, $i = 1, 2, 3$, into three intervals. The stabilizing feedback gains are obtained by solving Theorem 1, with

$$X_1 = \begin{bmatrix} 42.30 & -9.09 \\ -9.09 & 7.68 \end{bmatrix},$$

$$U_1 = 10^3 \times \begin{bmatrix} 1.78 & 0.02 & -0.92 & -1.23 \\ 0.02 & 1.72 & -1.04 & -1.12 \\ -0.92 & -1.04 & 3.97 & 4.11 \\ -1.23 & -1.12 & 4.11 & 5.77 \end{bmatrix},$$

$$U_2 = 10^3 \times \begin{bmatrix} 1.38 & -0.26 & -0.06 & -0.41 \\ -0.26 & 1.34 & -0.23 & -0.29 \\ -0.06 & -0.23 & 1.18 & 0.31 \\ -0.41 & -0.29 & 0.31 & 2.18 \end{bmatrix},$$

$$U_3 = 10^3 \times \begin{bmatrix} 1.18 & -0.36 & 0.13 & -0.29 \\ -0.36 & 1.08 & -0.05 & -0.07 \\ 0.13 & -0.05 & 0.65 & -0.48 \\ -0.29 & -0.07 & -0.48 & 1.28 \end{bmatrix}.$$

The obtained feedback gains for switching control are also listed in Tab. B.1. This switching control approach is compared with the worst-case design approach in the simulations with the initial condition $\mathbf{x}(\theta) = [-2 \quad 1.5]^T$, $\theta \in [-s_3, 0]$. The probability distributions for different delay intervals are set to $p_1 = 0\%$, $p_2 = 0\%$, and $p_3 = 100\%$. The controller chosen for the worst-case approach is

$$K = [-5.36, -3.82].$$

The simulation is performed 50 times with different sampling paths of delays. The simulation results are shown in Fig. B.1. It is observed that the switching control approach converges to $||x(t)|| = 0.05$ after $t_{||x(t)||=0.05} = 0.77\,\mathrm{s}$, 31.3% faster than the worst-case design $t_{||x(t)||=0.05} = 1.12\,\mathrm{s}$.

Example 2 In Chapter 4, the cost-performance trade-off of NVSC system is investigated. In the this example, the optimal network utilization in terms of occurrence probabilities of sending intervals is given.

The NVSC system in (B.1) is also considered in this example. The boundaries of the transmission delay, computation delay, and sending interval are the same as in Example 1. Assume the sum of the transmission time and computation time delays is categorized into two intervals by s_i^{c+x}, $i = 1, 2$:

$$\begin{aligned} s_1^{c+x} &= 25\,\mathrm{ms}, \quad & p_1^{c+x} &= 70\%, \\ s_2^{c+x} &= 45\,\mathrm{ms}, \quad & p_2^{c+x} &= 30\%, \end{aligned}$$

where p_i^{c+x}, $i = 1, 2$, denotes the occurrence probability of the associated sampling interval. Consider the image data is sampled and sent over the network with sampling interval d_k, which is further categorized into two intervals by $s_1^d = 10\,\mathrm{ms}$, and $s_2^d = 30\,\mathrm{ms}$. According to (4.13), the overall time delay is categorized into three intervals

$$\begin{aligned} s_1 &= s_1^{c+x} + s_1^d = 35\,\mathrm{ms}, & p_1 &= 0.7p_1^d, \\ s_2 &= s_1^{c+x} + s_2^d = s_2^{c+x} + s_1^d = 55\,\mathrm{ms}, & p_2 &= 0.7p_2^d + 0.3p_1^d, \\ s_3 &= s_2^{c+x} + s_2^d = 75\,\mathrm{ms}, & p_3 &= 0.3p_2^d. \end{aligned}$$

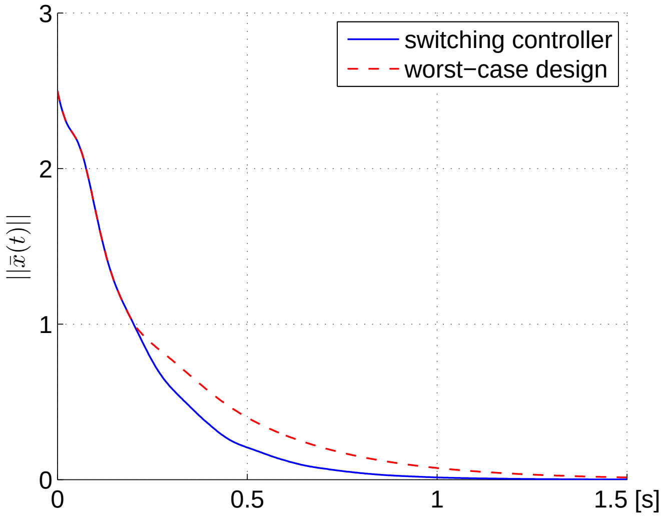


Figure B.1: The state trajectories $||x(t)||$ of NVSC with switching controller (solid line) and NVSC with worst-case design approach (dashed line).

According to Proposition 1, the design objective is to determine the optimal occurrence probability distribution of different sending intervals p_1^d, p_2^d such that he optimal network utilization is achieved; meanwhile, the guaranteed control performance is ensured. The network factors associated to different sending interval are $C_1 = 4 \times 10^4$, and $C_2 = 1 \times 10^4$, chosen heuristically. Set the parameter $R = \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{bmatrix}$. Then, the optimization function *fmincon* and the *Yalmip* is applied to solve the optimization problem in Proposition 1. With the initial condition $x(\theta) = [-2 \ 1.5]^T$, $\theta \in [-s_3, 0]$, the optimal occurrence probabilities are obtained

$$p_1^d = 37.45\%, \quad p_2^d = 62.55\%,$$

for $J = 1.63 \times 10^3$. And the associated stabilizing control feedback gains are

$$K_1 = [-17.29 \ -24.94], \quad K_2 = [-4.95 \ -7.12], \quad K_3 = [-2.10 \ -3.01],$$

where

$$X_1 = \begin{bmatrix} 0.60 & 0.01 \\ 0.01 & 0.18 \end{bmatrix},$$

$$U_1 = 10^3 \times \begin{bmatrix} 0.92 & -0.63 & -0.97 & 0.54 \\ -0.63 & 0.70 & 0.71 & -0.58 \\ -0.97 & 0.71 & 1.05 & -0.59 \\ 0.54 & -0.58 & -0.59 & 0.50 \end{bmatrix},$$

$$U_2 = \begin{bmatrix} 335.39 & -158.72 & -337.63 & 153.56 \\ -158.74 & 207.36 & 178.41 & -174.22 \\ -337.63 & 178.41 & 352.30 & -159.84 \\ 153.56 & -174.22 & -159.84 & 166.05 \end{bmatrix},$$

$$U_3 = \begin{bmatrix} 429.30 & -293.83 & -432.07 & 287.07 \\ -293.83 & 251.72 & 311.37 & -222.51 \\ -432.07 & 311.37 & 444.07 & -294.64 \\ 287.07 & -222.51 & -294.64 & 211.86 \end{bmatrix},$$

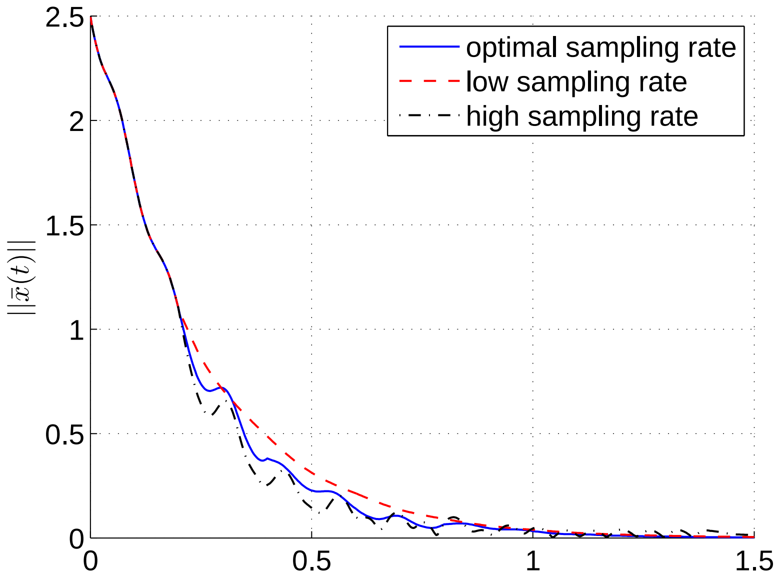


Figure B.2: The trajectories of th NVSC with optimal sending rate (solid line), high sending rate (dashed-dot line), and low sending rate (dashed line).

The NVSC system with the obtained controller and the sending rate scheduling strategy with optimal occurrence probabilities are demonstrated in the simulations. For comparison, two non-scheduling NVSC systems with high or low sending rate are simulated

$$\text{high sampling rate: } p_1^d = 100\%, \quad p_2^d = 0\%;$$
$$\text{low sampling rate: } p_1^d = 0\%, \quad p_2^d = 100\%.$$

The simulation results are shown in Fig. B.2. On one side, the trajectory of the NVSC with optimal sending rate converges to $||x|| = 0.05$ after $t_{||x||=0.05} = 0.716\,\text{s}$, 23.7% faster than the low sending rate approach, see Tab. B.2. On the other side, the networked usage is effectively reduced, 31.2% less than the approach with high sending rate. The results demonstrate the superior performance at less network load is achieved by the NVSC system with optimal sending rate.

Table B.2: Control performance and network load.

	$t_{\|\|x\|\|=0.05}$ [s]	Network load
optimal sampling rate	0.761	1.376×10^3
high sampling rate	0.652	2×10^3
low sampling rate	0.938	1×10^3

B.2 Network Emulator

Network emulator is an important tool for the research disclosed to networks. Since the network behavior is hard to be reproduced, it is desirable to use the network emulator

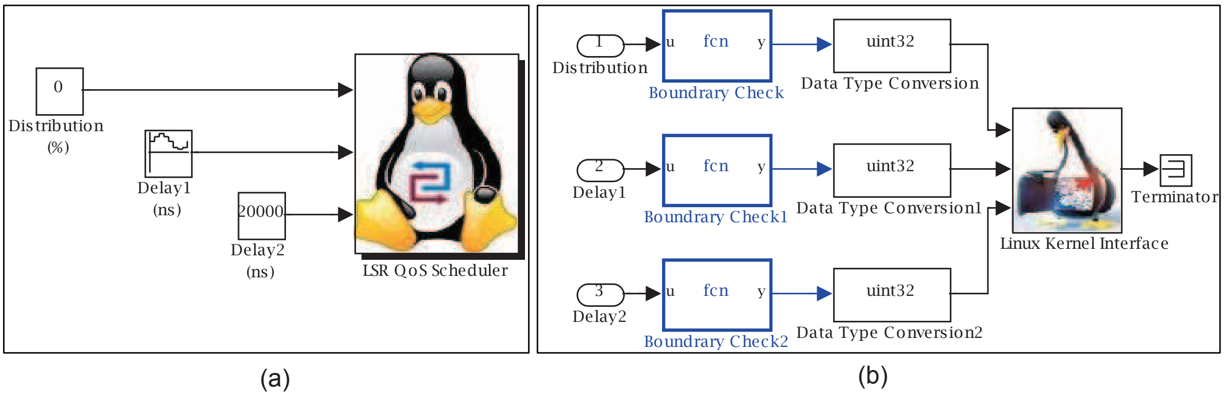


Figure B.3: Linux QoS scheduler encapsulated in Simulink block (a) and netlink interface in Simulink (b).

which emulates the network performance in a controlled environment. In the experiment of performance evaluation with the switching controller in Chapter 4, the random transmission delay caused by image data transmission over the network is simulated by a Linux kernel based network emulator NetEm [231] in this thesis.

NetEm is one of the post popular network emulators, and is built using existing Quality of Service (QoS) and differentiated Services facilities in the Linux kernel. It provides emulation of the network features such as variable delay, jitter, loss and so on. NetEm comprises of two parts namely the kernel module and the command line. The former is in charges of queuing discipline (qdisc), while the latter is for parameter configuration. The queuing discipline used by NetEm is First-In-First-out (FIFO) queuing between the protocol output and the network device. The communication between the command line and the Linux kernel is realized through the Netlink socket interface [232].

A SIMULINK wrapper shown in Fig. B.3 is implemented to access qdisc in Linux kernel through Netlink based on s-function of SIMULINK. The parameters could be changed under the real-time constraints. With the NetEm introduced above, the varying transmission delay in a LAN can be emulated for the experiments.

Bibliography

[1] P. Corke, “Visual control of robot manipulators – a review,” in *Visual Servoing*, pp. 1–31, World Scientific, 1994.

[2] S. Hutchinson, G. Hager, and P. Corke, “A tutorial on visual servo control,” *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 651–670, 1996.

[3] P. Corke, *Visual Control of Robots: high-performance visual servoing.* Research Studies Press, 1996.

[4] N. Gans, S. Hutchinson, and P. Corke, “Performance tests for visual servo control systems, with application to partitioned approaches to visual servo control,” *The International Journal of Robotics Research*, vol. 22, no. 10-11, p. 955, 2003.

[5] S. Funiak, C. Guestrin, M. Paskin, and R. Sukthankar, “Distributed localization of networked cameras,” in *Proc. of the 5th inter. conf. on Information processing in sensor networks*, pp. 34–42, ACM, 2006.

[6] B. Wilburn, N. Joshi, V. Vaish, E. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, “High performance imaging using large camera arrays,” *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3, pp. 765–776, 2005.

[7] B. Weber and K. Kühnlenz, “Visual servoing using triangulation with an omnidirectional multi-camera system,” in *Proc. of the International Conference on Control, Automation, Robotics and Vision (ICARCV)*, pp. 1440–1445, 2010.

[8] F. Chaumette, “Image moments: a general and useful set of features for visual servoing,” *IEEE Trans. Robot.*, vol. 20, no. 4, pp. 713–723, 2004.

[9] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *Inter. journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[10] M. Nixon and A. Aguado, *Feature extraction and image processing.* Academic Press, 2008.

[11] V. Sunderam *et al.*, “PVM: A framework for parallel distributed computing,” *Concurrency Practice and Experience*, vol. 2, no. 4, pp. 315–339, 1990.

[12] D. Henrich and T. Honiger, “Parallel processing approaches in robotics,” in *Proc. IEEE International Symp. Indus. Elec. (ISIE’97)*, vol. 2, pp. 702–707, 1997.

[13] R. Buyya, “High performance cluster computing: Architectures and systems, volume 1,” *Prentice Hall PTR*, vol. 82, pp. 327–350, 1999.

[14] X. Li, B. Veeravalli, and C. Ko, “Distributed image processing on a network of workstations,” *International Journal of Computers and Applications*, vol. 25, no. 2, pp. 136–145, 2003.

[15] Brost, A. and Haag, J., “Neural networks in the cockpit of the fly,” *Journal of Comparative Physiology A*, vol. 188, pp. 419–437, 2002.

[16] J. Haag and A. Borst, “Neural mechanism underlying complex receptive field properties of motion-sensitive interneurons,” *Nature neuroscience*, vol. 7, no. 6, pp. 628–634, 2004.

[17] R. Sreeja, D. Carroll, and A. Derek, “Man-made velocity estimators based on insect vision,” *Smart Materials and Structures*, vol. 14, no. 2, pp. 413–424, 2005.

[18] T. Bernard, B. Zavidovique, and F. Devos, “A programmable artificial retina,” *IEEE Journal of Solid-State Circuits*, vol. 28, no. 7, pp. 789–798, 1993.

[19] I. Ishii, Y. Nakabo, and M. Ishikawa, “Target tracking algorithm for 1 ms visual feedback system using massively parallel processing,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 3, pp. 2309–2314, IEEE, 1996.

[20] J. Eklund, C. Svensson, and A. Astrom, “VLSI implementation of a focal plane image processor-a realization of the near-sensor image processing concept,” *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 4, no. 3, pp. 322–335, 1996.

[21] T. Komuro, S. Kagami, and M. Ishikawa, “A dynamically reconfigurable SIMD processor for a vision chip,” *IEEE Journal of Solid-State Circuits*, vol. 39, no. 1, pp. 265–268, 2004.

[22] Y. Nakabo, M. Ishikawa, H. Toyoda, and S. Mizuno, “1 ms column parallel vision system and its application of high speed target tracking,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, vol. 1, pp. 650–655, IEEE, 2000.

[23] Y. Watanabe, T. Komuro, and M. Ishikawa, “955-fps real-time shape measurement of a moving/deforming object using high-speed vision for numerous-point analysis,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3192–3197, IEEE, 2007.

[24] J. Díaz, E. Ros, F. Pelayo, E. Ortigosa, and S. Mota, “FPGA-based real-time optical-flow system,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 2, pp. 274–279, 2006.

[25] I. Ishii, T. Taniguchi, R. Sukenobe, and K. Yamamoto, “Development of high-speed and real-time vision platform, H^3 vision,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3671–3678, IEEE, 2009.

[26] N. Furukawa, A. Namiki, S. Taku, and M. Ishikawa, "Dynamic regrasping using a high-speed multifingered hand and a high-speed vision system," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 181–187, IEEE, 2006.

[27] K. Shimizu and S. Hirai, "CMOS+ FPGA vision system for visual feedback of mechanical systems," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2060–2065, IEEE, 2006.

[28] R. Pieters, P. Jonker, and H. Nijmeijer, "High performance visual servoing for controlled μm-positioning," in *Proceedings of the 8th World Congress on Intelligent Control and Automation (WCICA)*, pp. 379–384, IEEE, 2010.

[29] J. Owens, D. Luebke, N. Govindaraju, M. Harris, J. Krüger, A. Lefohn, and T. Purcell, "A survey of general-purpose computation on graphics hardware," in *Computer graphics forum*, vol. 26, pp. 80–113, Wiley Online Library, 2007.

[30] M. Hopf and T. Ertl, "Hardware based wavelet transformations," in *Vision, Modeling, and Visualization*, vol. 99, pp. 317–328, 1999.

[31] R. Yang, M. Pollefeys, H. Yang, and G. Welch, "A unified approach to real-time, multi-resolution, multi-baseline 2d view synthesis and 3d depth estimation using commodity graphics hardware," *International Journal of Image and Graphics*, vol. 4, no. 4, pp. 627–651, 2004.

[32] R. Strzodka and C. Garbe, "Real-time motion estimation and visualization on graphics cards," in *Proceedings of the conference on Visualization'04*, pp. 545–552, IEEE Computer Society, 2004.

[33] NVIDIA Corporation, "NVIDIA CUDA unified device architecture." `http://www.nvidia.com/object/cuda_home_new.html/`, July,2008.

[34] B. Horn and B. Schunck, "Determining optical flow," *Artificial intelligence*, vol. 17, no. 1-3, pp. 185–203, 1981.

[35] Y. Mizukami and K. Tadamura, "Optical flow computation on compute unified device architecture," in *Proceedings of the International Conference on Image Analysis and Processing*, pp. 179–184, IEEE, 2007.

[36] T. Xu, T. Pototschnig, K. Kühnlenz, and M. Buss, "A high-speed multi-GPU implementation of bottom-up attention using CUDA," in *Proc. IEEE Inter. Conf. Robot. Auto. (ICRA)*, pp. 41–47, 2009.

[37] D. Pham, S. Asano, M. Bolliger, M. Day, H. Hofstee, C. Johns, J. Kahle, A. Kameyama, J. Keaty, Y. Masubuchi, *et al.*, "The design and implementation of a first-generation CELL processor," in *Proceedings of the IEEE International Solid-State Circuits Conference*, pp. 184–592, Ieee, 2005.

[38] P. Kongetira, K. Aingaran, and K. Olukotun, "Niagara: A 32-way multithreaded sparc processor," *Micro, IEEE*, vol. 25, no. 2, pp. 21–29, 2005.

[39] L. Seiler, D. Carmean, E. Sprangle, T. Forsyth, M. Abrash, P. Dubey, S. Junkins, A. Lake, J. Sugerman, R. Cavin, *et al.*, "Larrabee: a many-core x86 architecture for visual computing," in *ACM Transactions on Graphics (TOG)*, vol. 27, p. 18, ACM, 2008.

[40] G. Blake, R. Dreslinski, and T. Mudge, "A survey of multicore processors," *Signal Processing Magazine, IEEE*, vol. 26, no. 6, pp. 26–37, 2009.

[41] T. Chen, D. Budnikov, C. Hughes, and Y. Chen, "Computer vision on multi-core processors: Articulated body tracking," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 1862–1865, IEEE, 2007.

[42] T. Ike, N. Kishikawa, and B. Stenger, "A real-time hand gesture interface implemented on a multi-core processor," in *IAPR Conference on Machine Vision Applications*, pp. 9–12, Citeseer, 2007.

[43] Y. Chen, W. Li, J. Li, and T. Wang, "Novel parallel hough transform on multi-core processors," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pp. 1457–1460, IEEE, 2008.

[44] N. Gac, S. Mancini, M. Desvignes, and D. Houzet, "High speed 3D tomography on CPU, GPU, and FPGA," *EURASIP Journal on Embedded systems*, vol. 2008, p. 5, 2008.

[45] M. Smelyanskiy, D. Holmes, J. Chhugani, A. Larson, D. Carmean, D. Hanson, P. Dubey, K. Augustine, D. Kim, A. Kyker, *et al.*, "Mapping high-fidelity volume rendering for medical imaging to CPU, GPU and many-core architectures," *IEEE Transactions on Visualization and Computer Graphics*, vol. 15, no. 6, pp. 1563–1570, 2009.

[46] V. Lee, C. Kim, J. Chhugani, M. Deisher, D. Kim, A. Nguyen, N. Satish, M. Smelyanskiy, S. Chennupaty, P. Hammarlund, *et al.*, "Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU," in *ACM SIGARCH Computer Architecture News*, vol. 38, pp. 451–460, ACM, 2010.

[47] P. Benner, P. Ezzatti, E. Quintana-Ortí, and A. Remón, "Using hybrid CPU-GPU platforms to accelerate the computation of the matrix sign function," in *Euro-Par 2009–Parallel Processing Workshops*, pp. 132–139, Springer, 2010.

[48] M. Daga, A. Aji, and W. Feng, "On the efficacy of a fused CPU+ GPU processor (or APU) for parallel computing," in *2011 Symposium on Application Accelerators in High-Performance Computing (SAAHPC)*, pp. 141–149, IEEE, 2011.

[49] "The top500 supercomputer sites." `http://www.top500.org`.

[50] G. Geist and V. Sunderam, "Network-based concurrent computing on the PVM system," *Concurrency: Practice and Experience*, vol. 4, pp. 293–312, 1992.

[51] E. Gabriel, M. Resch, T. Beisel, and R. Keller, "Distributed computing in a heterogeneous computing environment," *Recent Advances in Parallel Virtual Machine and Message Passing Interface*, pp. 180–187, 1998.

[52] C. Lee and M. Hamdi, "Parallel image processing applications on a network of workstations," *Parallel Computing*, vol. 21, no. 1, pp. 137–160, 1995.

[53] S. Ranka and S. Sahni, "Image template matching on mimd hypercube multicomputers," *Journal of Parallel and Distributed Computing*, vol. 10, no. 1, pp. 79–84, 1990.

[54] I. Abdou and W. Pratt, "Quantitative design and evaluation of enhancement/thresholding edge detectors," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 753–763, 1979.

[55] K. Hawick, P. Coddington, and H. James, "Distributed frameworks and parallel algorithms for processing large-scale geographic data," *Parallel Computing*, vol. 29, no. 10, pp. 1297–1333, 2003.

[56] J. Graham, "Special computer architectures for robotics: Tutorial and survey," *IEEE Transactions on and Robotics and Automation*, vol. 5, no. 5, pp. 543–554, 1989.

[57] R. Tong, R. Wishner, and E. Tse, "Distributed hypothesis formation in sensor fusion systems," in *Decision and Control including the Symposium on Adaptive Processes, 1981 20th IEEE Conference on*, vol. 20, pp. 1421–1424, IEEE, 1981.

[58] V. Gylys and J. Edwards, "Optimal partitioning of workload for distributed systems," in *Digest COMPCON Fall*, pp. 353–357, 1976.

[59] S. Bokhari, "Partitioning problems in parallel, pipeline, and distributed computing," *IEEE Transactions on Computers*, vol. 37, no. 1, pp. 48–57, 1988.

[60] M. Gupta and P. Banerjee, "Compile-time estimation of communication costs on multicomputers," in *Proceedings of the 6th International Parallel Processing Symposium*, pp. 470–475, IEEE, 1992.

[61] V. Bharadwaj, *Scheduling divisible loads in parallel and distributed systems.* Wiley-IEEE Computer Society Pr, 1996.

[62] F. Zhao and L. Guibas, *Wireless sensor networks: an information processing approach.* Morgan Kaufmann Pub, 2004.

[63] H. Wu and A. Abouzeid, "Energy efficient distributed jpeg2000 image compression in multihop wireless networks," *Proceedings of the 4thWorkshop on Applications and Services in Wireless Networks (ASWN04), Boston, Mass, USA*, pp. 152–160, 2004.

[64] R. Wagner, R. Nowak, and R. Baraniuk, "Distributed image compression for sensor networks using correspondence analysis and super-resolution," in *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, vol. 1, pp. I–597, IEEE, 2003.

[65] K. Lui and E. Lam, "Image transmission in sensor networks," in *Signal Processing Systems Design and Implementation, 2005. IEEE Workshop on*, pp. 726–730, IEEE, 2005.

[66] K. Chow, K. Lui, and E. Lam, “Efficient on-demand image transmission in visual sensor networks,” *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 225–225, 2007.

[67] V. Lecuire, C. Duran-Faundez, and N. Krommenacker, “Energy-efficient image transmission in sensor networks,” *International Journal of Sensor Networks*, vol. 4, no. 1, pp. 37–47, 2008.

[68] M. Ding, X. Cheng, and G. Xue, “Aggregation tree construction in sensor networks,” in *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, vol. 4, pp. 2168–2172, IEEE, 2003.

[69] N. Shrivastava, C. Buragohain, D. Agrawal, and S. Suri, “Medians and beyond: new aggregation techniques for sensor networks,” in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 239–249, ACM, 2004.

[70] N. Gehrig and P. Dragotti, “Distributed compression in camera sensor networks,” in *IEEE 6th Workshop on Multimedia Signal Processing*, pp. 311–314, IEEE, 2004.

[71] M. Wu and C. Chen, “Collaborative image coding and transmission over wireless sensor networks,” *EURASIP Journal on Applied Signal Processing*, vol. 2007, no. 1, pp. 223–223, 2007.

[72] J. Danskin, G. Davis, and X. Song, “Fast lossy internet image transmission,” in *Proceedings of the Third ACM Conference on Multimedia, San Francisco, California*, pp. 321–332, 1995.

[73] T. Wang, H. Fang, and L. Chen, “Low-delay and error-robust wireless video transmission for video communications,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 12, no. 12, pp. 1049–1058, 2002.

[74] W. Tan and A. Zakhor, “Real-time internet video using error resilient scalable compression and tcp-friendly transport protocol,” *Multimedia, IEEE Transactions on*, vol. 1, no. 2, pp. 172–186, 1999.

[75] D. Wu, Y. Hou, and Y. Zhang, “Transporting real-time video over the internet: Challenges and approaches,” *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1855–1877, 2000.

[76] D. Jansen and H. Buttner, “Real-time Ethernet: the EtherCAT solution,” *Computing and Control Engineering*, vol. 15, pp. 16–21, 2004.

[77] R. Davis, A. Burns, R. Bril, and J. Lukkien, “Controller area network (CAN) schedulability analysis: Refuted, revisited and revised,” *Real-Time Systems*, vol. 35, no. 3, pp. 239–272, 2007.

[78] `http://www.sercos.com/`.

[79] R. Jordan and C. Abdallah, “Wireless communications and networking: an overview,” *Antennas and Propagation Magazine, IEEE*, vol. 44, no. 1, pp. 185–193, 2002.

[80] D. Kragic and H. Christensen, "Survey on visual servoing for manipulation," *Computational Vision and Active Perception Laboratory, Fiskartorpsv*, vol. 15, 2002.

[81] F. Chaumette and S. Hutchinson, "Visual servo control. I. Basic approaches," *Robotics & Automation Magazine, IEEE*, vol. 13, no. 4, pp. 82–90, 2006.

[82] L. Weiss, A. Sanderson, and C. Neuman, "Dynamic visual servo control of robots: an adaptive image-based approach," in *Robotics and Automation. Proceedings. 1985 IEEE International Conference on*, vol. 2, pp. 662–668, IEEE, 1985.

[83] K. Deguchi, "Optimal motion control for image-based visual servoing by decoupling translation and rotation," in *Intelligent Robots and Systems, 1998. Proceedings., 1998 IEEE/RSJ International Conference on*, vol. 2, pp. 705–711, IEEE, 1998.

[84] E. Malis, F. Chaumette, and S. Boudet, "21/2d visual servoing," *Robotics and Automation, IEEE Transactions on*, vol. 15, no. 2, pp. 238–250, 1999.

[85] P. Corke and S. Hutchinson, "A new partitioned approach to image-based visual servo control," *Robotics and Automation, IEEE Transactions on*, vol. 17, no. 4, pp. 507–515, 2001.

[86] F. Chaumette, S. Hutchinson, *et al.*, "Visual servo control, part ii: Advanced approaches," *IEEE Robotics and Automation Magazine*, vol. 14, no. 1, pp. 109–118, 2007.

[87] N. Gans and S. Hutchinson, "An asymptotically stable switched system visual controller for eye in hand robots," in *Proceedings. 2003 IEEE/RSJ International Conference o Intelligent Robots and Systems (IROS)*, vol. 1, pp. 735–742, IEEE, 2003.

[88] `http://www.ptgrey.com/`.

[89] T. Zhang, W. Li, K. Kuhnlenz, and M. Buss, "Multi-sensory motion estimation and control of an autonomous quadrotor," *Advanced Robotics*, vol. 25, no. 11, pp. 1493–1514, 2011.

[90] `http://www.microtron.be/`.

[91] T. Zhang, H. Wu, A. Borst, K. Kühnlenz, and M. Buss, "An FPGA implementation of insect-inspired motion detector for high-speed vision systems," in *Prod. IEEE Inter. Conf. Robot. Auto. (ICRA)*, pp. 335–340, 2008.

[92] `http://www.xbox.com/en-US/kinect`.

[93] D. Lazewatsky and W. Smart, "An inexpensive robot platform for teleoperation and experimentation," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 1211–1216, IEEE, 2011.

[94] D. Schuurman and D. Capson, "Robust direct visual servo using network-synchronized cameras," *Robotics and Automation, IEEE Transactions on*, vol. 20, no. 2, pp. 319–334, 2004.

[95] V. Vaish, B. Wilburn, N. Joshi, and M. Levoy, “Using plane+ parallax for calibrating dense camera arrays,” in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. I–2, IEEE, 2004.

[96] C. Zhang and T. Chen, “A self-reconfigurable camera array,” in *ACM SIGGRAPH 2004 Sketches*, p. 151, ACM, 2004.

[97] T. Drummond and R. Cipolla, “Real-time visual tracking of complex structures,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 932–946, 2002.

[98] C. Kervrann and F. Heitz, “A hierarchical Markov modeling approach for the segmentation and tracking of deformable shapes,” *Graphical Models and Image Processing*, vol. 60, no. 3, pp. 173–195, 1998.

[99] A. Comport, É. Marchand, and F. Chaumette, “Robust model-based tracking for robot vision,” in *Intelligent Robots and Systems, 2004.(IROS 2004). Proceedings. 2004 IEEE/RSJ International Conference on*, vol. 1, pp. 692–697, IEEE, 2004.

[100] P. Scovanner, S. Ali, and M. Shah, “A 3-dimensional SIFT descriptor and its application to action recognition,” in *Proceedings of the 15th international conference on Multimedia*, pp. 357–360, ACM, 2007.

[101] H. Zhou, Y. Yuan, and C. Shi, “Object tracking using SIFT features and mean shift,” *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 345–352, 2009.

[102] T. Xu, T. Zhang, K. Kühnlenz, and M. Buss, “Attentional object detection with an active multi-focal vision system,” *International Journal of Humanoid Robotics*, vol. 7, no. 2, pp. 223–243, 2010.

[103] H. Bay, T. Tuytelaars, and L. Van Gool, “Surf: Speeded up robust features,” *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[104] W. Chen, Y. Xiong, J. Gao, N. Gelfand, and R. Grzeszczuk, “Efficient extraction of robust image features on mobile devices,” in *Proceedings of the 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality*, pp. 1–2, IEEE Computer Society, 2007.

[105] A. Ess, B. Leibe, K. Schindler, and L. Van Gool, “A mobile vision system for robust multi-person tracking,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1–8, IEEE, 2008.

[106] A. Murillo, J. Guerrero, and C. Sagues, “Surf features for efficient robot localization with omnidirectional images,” in *Robotics and Automation, 2007 IEEE International Conference on*, pp. 3901–3907, IEEE, 2007.

[107] S. Sinha, J. Frahm, M. Pollefeys, and Y. Genc, “Gpu-based video feature tracking and matching,” in *EDGE, Workshop on Edge Computing Using New Commodity Architectures*, vol. 278, Citeseer, 2006.

[108] T. Terriberry, L. French, and J. Helmsen, "Gpu accelerating speeded-up robust features," in *Proc of 3DPVT*, Citeseer, 2008.

[109] D. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 2, pp. 1150–1157, Ieee, 1999.

[110] T. Tuytelaars and K. Mikolajczyk, "Local invariant feature detectors: a survey," *Foundations and Trends® in Computer Graphics and Vision*, vol. 3, no. 3, pp. 177–280, 2008.

[111] R. Kelly, "Robust asymptotically stable visual servoing of planar robots," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 5, pp. 759–766, 1996.

[112] L. Hsu, R. Costa, and P. Aquino, "Stable adaptive visual servoing for moving targets," in *American Control Conference, 2000. Proceedings of the 2000*, vol. 3, pp. 2008–2012, IEEE, 2000.

[113] P. Corke and M. Good, "Dynamic effects in visual closed-loop systems," *IEEE Trans. Robot. Automat.*, vol. 12, no. 5, pp. 671–683, 1996.

[114] J. Zhang, R. Lumia, J. Wood, and G. Starr, "Delay dependent stability limits in high performance real-time visual servoing systems," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, vol. 1, pp. 485–491, IEEE, 2003.

[115] G. Welch and G. Bishop, "An introduction to the kalman filter," *Design*, vol. 7, no. 1, pp. 1–16, 2001.

[116] W. Wilson, C. Williams Hulls, and G. Bell, "Relative end-effector control using cartesian position based visual servoing," *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 5, pp. 684–696, 1996.

[117] D. Clarke, C. Mohtadi, and P. Tuffs, "Generalized predictive control–part i. the basic algorithm," *Automatica*, vol. 23, no. 2, pp. 137–148, 1987.

[118] J. Gangloff, M. De Mathelin, and G. Abba, "6 dof high speed dynamic visual servoing using gpc controllers," in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 3, pp. 2008–2013, IEEE, 1998.

[119] K. Shin and X. Cui, "Computing time delay and its effects on real-time control systems," *IEEE Transactions on Control Systems Technology*, vol. 3, no. 2, pp. 218–224, 1995.

[120] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[121] J. Baillieul and P. Antsaklis, "Control and communication challenges in networked real-time systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 9–28, 2007.

[122] T. Chen and B. Francis, *Optimal Sampled-Data Control Systems.* Berlin: Springer-Verlag, 1995.

[123] F. Yang, Z. Wang, Y. Hung, and M. Gani, "H_∞ control for networked systems with random communication delays," *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 511–518, 2006.

[124] L. Xiao, A. Hassibi, and J. P. How, "Control with random communication delays via a discrete-time jump system approach," in *Proceedings of the American Control Conference*, (Chicago, Illinois), June 2000.

[125] L. Zhang, Y. Shi, T. Chen, and B. Huang, "A new method for stabilization of networked control systems with random delays," *IEEE Trans. Automat. Control*, vol. 20, no. 8, pp. 117–1181, 2005.

[126] C. Chen, S. Hirche, and M. Buss, "Controller design and experimental validation for networked control systems with time-varying random delay," *Journal of the Society of Instrument and Control Engineers*, vol. 47, no. 8, pp. 676–685, 2008.

[127] S. Hirche, C.-C. Chen, and M. Buss, "Performance oriented control over networks: switching controllers and switched time delay," *Asian Journal of Control*, pp. 24–33, Jan. 2008.

[128] C. Chen, A. Molin, and S. Hirche, "Guaranteed cost control over quality-of-service networks," in *Proc. of the European Control Conference, Budapest, Hungary*, pp. 2355–2360, 2009.

[129] C. Chen, S. Hirche, and M. Buss, "Stability, stabilization and experiments for networked control systems with random time delay," in *Proc. of the American Control Conference*, pp. 1552–1557, 2008.

[130] C. Chen, S. Hirche, and M. Buss, "Sampled-data networked control systems with random time delay," in *Proc. the 17th World Congress the International Federation of Automatic Control*, pp. 11594–11599, 2008.

[131] S. Hirche, C. Chen, and M. Buss, "Performance oriented control over networks - switching controllers and switched time delay," in *Proc. 45th IEEE Conf. Decision and Control, San Diego*, pp. 4999–5005, 2006.

[132] C. Chen, S. Hirche, and M. Buss, "Towards quality-of-service control of networked control systems: a switched time delay systems approach," in *Proc. IEEE Inter. Conf. on Cont. Appl. (CCA)*, pp. 1325–1330, 2006.

[133] R. Gupta and M. Chow, "Networked control system: overview and research trends," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 7, pp. 2527–2535, 2010.

[134] Y. Tipsuwan and M. Chow, "Control methodologies in networked control systems," *Control engineering practice*, vol. 11, no. 10, pp. 1099–1111, 2003.

[135] T. Yang, "Networked control system: a brief survey," in *IEE Proc. -Control Theory Appl.*, vol. 153, pp. 403–412, IET, 2006.

[136] L. Litz, T. Gabriel, M. Groß, and O. Gabel, "Networked control systems (ncs)-stand und ausblick (networked control systems (ncs)-state of the art and future)," *at-Automatisierungstechnik*, vol. 56, no. 1/2008, pp. 4–19, 2008.

[137] M. Gaid, A. Çela, and Y. Hamam, "Optimal integrated control and scheduling of networked control systems with communication constraints: Application to a car suspension system," *IEEE Transactions on Control Systems Technology*, vol. 14, no. 4, pp. 776–787, 2006.

[138] P. Martí, J. Yépez, M. Velasco, R. Villà, and J. Fuertes, "Managing quality-of-control in network-based control systems by controller and message scheduling co-design," *Industrial Electronics, IEEE Transactions on*, vol. 51, no. 6, pp. 1159–1167, 2004.

[139] Y. Zhao, G. Liu, and D. Rees, "Integrated predictive control and scheduling co-design for networked control systems," *Control Theory & Applications, IET*, vol. 2, no. 1, pp. 7–15, 2008.

[140] P. Otanez, J. Moyne, and D. Tilbury, "Using deadbands to reduce communication in networked control systems," in *Proceedings of the American Control Conference*, vol. 4, pp. 3015–3020, 2002.

[141] S. Hirche, P. Hinterseer, E. Steinbach, and M. Buss, "Towards deadband control in networked teleoperation systems," in *Proc. Int. Fed. Autom. Control (IFAC), Prague, Czech Republic*, 2005.

[142] K. Åström and B. Bernhardsson, "Comparison of Riemann and Lebesgue sampling for first order stochastic systems," in *Proc. of the 41st IEEE Conference on Decision and Control*, vol. 2, pp. 2011–2016, 2002.

[143] A. Molin and S. Hirche, "On LQG joint optimal scheduling and control under communication constraints," in *Proc. of the 48th IEEE Conf. on Decision and Control (CDC)*, pp. 5832–5838, 2009.

[144] A. Molin and S. Hirche, "Structural characterization of optimal event-based controllers for linear stochastic systems," in *Proc. of the 49th IEEE Conf. on Decision and Control (CDC)*, pp. 3227–3233, 2010.

[145] A. Molin and S. Hirche, "On the optimality of certainty equivalence for event-triggered control systems," *IEEE Transactions on Automatic Control*, (2012), to appear.

[146] K. Ji and W. Kim, "Stochastic optimal control and network co-design for networked control systems," *International Journal of Control Automation and Systems*, vol. 5, no. 5, p. 515, 2007.

[147] O. Imer, S. Yüksel, and T. Basar, "Optimal control of lti systems over unreliable communication links," *Automatica*, vol. 42, no. 9, pp. 1429–1439, 2006.

[148] M. Tabbara and D. Nesic, "Input–output stability of networked control systems with stochastic protocols and channels," *Automatic Control, IEEE Transactions on*, vol. 53, no. 5, pp. 1160–1175, 2008.

[149] X. Liu and A. Goldsmith, "Wireless network design for distributed control," in *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, vol. 3, pp. 2823–2829, IEEE, 2004.

[150] L. Xiao, M. Johansson, H. Hindi, S. Boyd, and A. Goldsmith, “Joint optimization of communication rates and linear systems,” *Automatic Control, IEEE Transactions on*, vol. 48, no. 1, pp. 148–153, 2003.

[151] B. Hassenstein and W. Reichardt, “Systemtheoretische analyse der zeit-, reihenfolgen-und vorzeichenauswertung bei der bewegungsperzeption des r sselk fers chlorophanus,” *ZNaturforsch*, vol. 11, no. b, pp. 513–524, 1956.

[152] H. Krapp, R. Hengstenberg, *et al.*, “Estimation of self-motion by optic flow processing in single visual interneurons,” *Nature*, vol. 384, no. 6608, pp. 463–466, 1996.

[153] H. Cuntz, J. Haag, F. Forstner, I. Segev, and A. Borst, “Robust coding of flow-field parameters by axo-axonal gap junctions between fly visual interneurons,” *Proceedings of the National Academy of Sciences*, vol. 104, no. 24, p. 10229, 2007.

[154] T. Poggio and W. Reichardt, “Visual control of orientation behaviour in the fly: Part ii. towards the underlying neural interactions,” *Quarterly Reviews of Biophysics*, vol. 9, no. 03, pp. 377–438, 1976.

[155] R. Harris and D. O’Carroll, “Afterimages in fly motion vision,” *Vision research*, vol. 42, no. 14, pp. 1701–1714, 2002.

[156] Rajesh, S., Carroll,D. O’, and Abbott, D., “Man-made velocity estimators based on insect vision,” *Smart Materials and Structures*, vol. 14, pp. 413–424, 2005.

[157] N. Franceschini, C. Blanes, and L. Oufar, “Passive, non-contact optical velocity sensor,” *Dossier technique ANVAR/DVAR*, vol. 51549, 1986.

[158] F. Ruffier, S. Viollet, S. Amic, and N. Franceschini, “Bio-inspired optical flow circuits for the visual guidance of micro air vehicles,” in *Circuits and Systems, 2003. ISCAS’03. Proceedings of the 2003 International Symposium on*, vol. 3, pp. III–846, IEEE, 2003.

[159] Shih-Chi Liu, “A neuromorphic aVLSI model of global motion processing in the fly,” *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, pp. 1458–1467, 2000.

[160] Harrison, R.R. and Koch, C., “A robust analog VLSI motion sensor based on the visual system of the fly,” *Autonomous Robots*, vol. 7, pp. 211–224, 1999.

[161] Harrison, R.R. and Koch, C., “A silicon implementation of the fly’s optomotor control system,” *Neural Computation*, vol. 12, pp. 2291–2304, 2000.

[162] R. Harrison, “A biologically inspired analog ic for visual collision detection,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 52, no. 11, pp. 2308–2318, 2005.

[163] E. Nakamura, S. Asami, T. Takahashi, and K. Sawada, “Real time parameter optimization for elementary motion detectors,” in *Image Processing, 2006 IEEE International Conference on*, pp. 1065–1068, IEEE, 2006.

[164] F. Aubépart, M. El Farji, and N. Franceschini, “FPGA implementation of elementary motion detectors for the visual guidance of micro-air-vehicles,” in *Industrial Electronics, 2004 IEEE International Symposium on*, vol. 1, pp. 71–76, IEEE, 2004.

[165] Nakamura, E., Asami, S., Takahashi, T., and Sawada, K., “Real time parameter optimization for elementary motion detectors,” *Image Processing, 2006 IEEE International Conference on*, pp. 1065–1068, 2006.

[166] Aubepart, F., El Farji, M., and Franceschini, N., “FPGA implementation of elementary motion detectors for the visual guidance of micro-air-vehicles,” *Industrial Electronics, 2004 IEEE International Symposium on*, vol. 1, pp. 71–76, 2004.

[167] I. Bouchrika and M. Nixon, “Model-based feature extraction for gait analysis and recognition,” *Computer Vision/Computer Graphics Collaboration Techniques*, pp. 150–160, 2007.

[168] H. Bay, T. Tuytelaars, and L. Van Gool, “SURF: Speeded up robust features,” *Computer Vision–ECCV 2006*, pp. 404–417, 2006.

[169] W. Stevens, *TCP/IP Illustrated: the protocols*, vol. 1. Addison-Wesley Professional, 1994.

[170] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, “RFC3550: RTP: A transport protocol for real-time applications,” *RFC Editor United States*, 2003.

[171] *http://www.gnu.org/software/ccrtp/.*

[172] W. Stevens, “TCP congestion control,” *Internet Engineering Task Force, RFC2581*, 1999.

[173] S. W. R. Wright, Gary R., *TCP/IP illustrated. Vol.2: The implementation.* Addison-Wesley Professional Computing Series, 1995.

[174] H. Schulzrinne, “Rtp profile for audio and video conferences with minimal control,” *Internet Engineering Task Force, RFC3551.*

[175] T. Lindeberg, *Scale-space theory in computer vision.* Springer, 1993.

[176] T. Lindeberg, “Scale-space: A framework for handling image structures at multiple scales,” *CERN EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH-REPORTS-CERN*, pp. 27–38, 1996.

[177] J. Sporring, L. Florack, M. Nielsen, and P. Johansen, *Gaussian scale-space theory.* Kluwer Academic Publishers, 1997.

[178] J. Beis and D. Lowe, “Shape indexing using approximate nearest-neighbour search in high-dimensional spaces,” in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pp. 1000–1006, IEEE, 1997.

[179] M. Fischler and R. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[180] A. Ansar and K. Daniilidis, “Linear pose estimation from points or lines,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 5, pp. 578–589, 2003.

[181] E. Marchand and F. Chaumette, “Virtual visual servoing: a framework for real-time augmented reality,” in *Computer Graphics Forum*, vol. 21, pp. 289–297, Wiley Online Library, 2002.

[182] L. Montestruque and P. Antsaklis, “Stability of model-based networked control systems with time-varying transmission times,” *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1562–1572, 2004.

[183] L. Sciavicco and B. Siciliano, *Modelling and control of robot manipulators.* Springer Verlag, 2000.

[184] E. Fridman, A. Seuret, and J.-P. Richard, “Robust sampled-data stabilization of linear systems: an input delay approach,” *Automatica*, vol. 40, pp. 1441–1446, 2004.

[185] C. Chen, H. Wu, K. Kühnlenz, and Hirche, “Switching control for a networked vision-based control system,” *Automatisierungstechnik*, vol. 59, pp. 124–133, 2010.

[186] C. Grinstead and J. Snell, *Introduction to probability.* Amer Mathematical Society, 1997.

[187] J. Lofberg, “YALMIP: A toolbox for modeling and optimization in MATLAB,” in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, pp. 284–289, IEEE, 2004.

[188] X. Mao, “Exponential stability of stochastic delay interval systems with markovian switching,” *Automatic Control, IEEE Transactions on*, vol. 47, pp. 1604–1612, Oct 2002.

[189] E.-K. Boukas and Z.-K. Liu, *Deterministic and Stochastic Time Delay Systems.* Boston: Birkhäuser, 2002.

[190] F. Zhang, *The Schur complement and its applications*, vol. 4. Springer, 2005.

[191] S. Ethier and T. Kurtz, *Markov processes: Characterization and convergence*, vol. 6. Wiley New York, 1986.

[192] B. Stanczyk, *Development and Control of an Anthropomorphic Telerobotic System.* PhD thesis, Technische Universität München, 2006.

[193] M. Muja and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration,” in *Proc. Inter. Conf. Comp. Visi. Theo. and Appl. (VISSAPP’09)*, pp. 331–340, INSTICC Press, 2009.

[194] M. A. Fischer and R. C. Bolles, “Random sampled consensus: A paradigm for modeling fitting with applications to image analysis and automated cartography,” in *Communications of the ACM*, vol. 2, pp. 381–395, 1981.

[195] S. Laughlin and D. Osorio, “Mechanisms for neural signal enhancement in the blowfly compound eye,” *Journal of experimental biology*, vol. 144, no. 1, p. 113, 1989.

[196] C. Gilbert, D. Penisten, and R. DeVoe, “Discrimination of visual motion from flicker by identified neurons in the medulla of the fleshfly sarcophaga bullata,” *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, vol. 168, no. 6, pp. 653–673, 1991.

[197] J. Douglass and N. Strausfeld, “Visual motion detection circuits in flies: peripheral motion computation by identified small-field retinotopic neurons,” *The Journal of neuroscience*, vol. 15, no. 8, p. 5596, 1995.

[198] Egelhaaf, M., Borst, A., and Reichardt, W., “Computational structure of a biological motion-detection system as revealed by local detector analysis in the fly’s nervous system,” *Journal of the Optical Society of America A (Optics and Image Science)*, vol. 6, pp. 1070–1087, 1989.

[199] B. Hassenstein and W. Reichardt, “Systemtheoretische analyse der zeit-reihenfolgen, und vorzeihenauswertung bei der bewegungsperzeption des ruesselkaefers,” *Naturforch*, vol. 11b, p. 1956, 1956.

[200] Faille, F., Borst, A., and Faerber,G., “Biologically inspired motion detector for video interpretation.” received by Biological Cybernetics on April 12, 2007.

[201] A. Warzecha and M. Egelhaaf, “Intrinsic properties of biological motion detectors prevent the optomotor control system from getting unstable,” *The Royal Society*, vol. 351, pp. 1579–1791, 1996.

[202] T.G. Zhang, H.Y. Wu, A. Borst, K. Kuehnlenz, and M. Buss, “An fpga implementation of insect-inspired motion detector for high-speed vision systems,” *IEEE International Conference on Robotics and Automation (ICRA)*, pp. 335–340, 2008.

[203] Borst, A., Flanagin, V.L., and Sompolinsky, H., “Adaptation without parameter change: dynamic gain control in motion detection,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, pp. 6172–6176, 2005.

[204] Srinivasan, M.V., Poteser, M., and Kral, K., “Motion detection in insect orientation and navigation,” *Vision Research*, vol. 39, pp. 2749–2766, 1999.

[205] Borst, A., Reisenman, C., and Haag, J., “Adaptation of response transients in fly motion vision. ii. model studies,” *Vision Research*, vol. 43, pp. 1309–1322, 2003.

[206] Egelhaaf, M. and Bosrt, A., “Transient and stead-state response properties of movement detectors,” *Optical Society of America*, vol. 6, pp. 116–127, 1988.

[207] K. Kirschfeld, “The visual system of musca: studies on optics, structure and function,” 1972.

[208] M. Egelhaaf and A. Borst, “Transient and steady-state response properties of movement detectors,” *JOSA A*, vol. 6, no. 1, pp. 116–127, 1989.

[209] R. Brinkworth and D. O’Carroll, “Robust models for optic flow coding in natural scenes inspired by insect biology,” *PLoS Computational Biology*, vol. 5, no. 11, p. e1000555, 2009.

[210] Dror, R.O., O’Carroll, D.C., and Laughlin, S.B., “Accuracy of velocity estimation by reichardt correlators,” *Journal of the Optical Society of America A (Optics, Image Science and Vision)*, vol. 18, pp. 231–252, 2001.

[211] Adelson, E.H. and Bergen, J.R., “Spatiotemporal energy models for the perception of motion,” *Journal of the Optical Society of America A (Optics and Image Science)*, vol. 2, pp. 284–299, 1984.

[212] G. Horridge, “A template theory to relate visual processing to digital circuitry,” *Proceedings of the Royal Society of London. B. Biological Sciences*, vol. 239, no. 1294, p. 17, 1990.

[213] J. Zanker, M. Srinivasan, and M. Egelhaaf, “Speed tuning in elementary motion detectors of the correlation type,” *Biological cybernetics*, vol. 80, no. 2, pp. 109–116, 1999.

[214] A. Straw, T. Rainsford, and D. O’Carroll, “Contrast sensitivity of insect motion detectors to natural images,” *Journal of Vision*, vol. 8, no. 3, 2008.

[215] R. Dror, D. O’Carroll, and S. Laughlin, “Accuracy of velocity estimation by Reichardt correlators,” *Journal of the Optical Society of America A*, vol. 18, no. 2, pp. 241–252, 2001.

[216] R. Dror, D. OCarroll, and S. Laughlin, “The role of natural image statistics in biological motion estimation,” in *Biologically Motivated Computer Vision*, pp. 509–533, Springer, 2000.

[217] D. Field, “Relations between the statistics of natural images and the response properties of cortical cells,” *Journal of the Optical Society of America A*, vol. 4, no. 12, pp. 2379–2394, 1987.

[218] D. Ruderman and W. Bialek, “Statistics of natural images: Scaling in the woods,” *Physical Review Letters*, vol. 73, no. 6, pp. 814–817, 1994.

[219] A. van der Schaaf and J. Van Hateren, “Modelling the power spectra of natural images: statistics and information,” *Vision Research*, vol. 36, no. 17, pp. 2759–2770, 1996.

[220] N. Nill, “Scene power spectra: the moment as an image quality merit factor,” *Applied Optics*, vol. 15, no. 11, pp. 2846–2854, 1976.

[221] S. Single and A. Borst, “Dendritic integration and its role in computing image velocity,” *Science*, vol. 281, pp. 1848–1850, 1998.

[222] R. Mahony, P. Corke, and T. Hamel, "Dynamic image-based visual servo control using centroid and optical flow freatures," *Dynamic Systems, Meaturement, and Control*, vol. 130, pp. 1–11, 2008.

[223] A. Crtual and F. Chaumette, "Positionnning a camera parallel to a plane using dynamic visual servoing," *Conf. on Intelligent Robotis and Systems*, vol. 1, pp. 43–48, 1997.

[224] B. Stanczyk, *Development and Control of an Anthropomorphic Telerobotic System.* PhD thesis, Technische Universität München, 2006.

[225] A. Warzecha and M. Egelhaaf, "Intrinsic properties of biological motion detectors prevent the optomotor control system from getting unstable," *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, vol. 351, no. 1347, pp. 1579–1591, 1996.

[226] R. Kelly, "Stable visual servoing of camera-in-hand robotic systems," *Transactions on Mechatronics*, vol. 5, pp. 39–48, 2000.

[227] G. Hager and P. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 10, pp. 1025–1039, 1998.

[228] A. Bruhn, J. Weickert, and C. Schnörr, "Lucas/kanade meets horn/schunck: Combining local and global optic flow methods," *International Journal of Computer Vision*, vol. 61, no. 3, pp. 211–231, 2005.

[229] Y. Nakamura, *Advanced robotics: redundancy and optimization.* Addison-Wesley Longman Publishing Co., Inc., 1990.

[230] `http://www.mikrotron.de`.

[231] "Net:netem." `http://www.linuxfoundation.org/en/Net:Netem`.

[232] "Netlink socket - overview." `http://qos.ittc.ku.edu/netlink/html/index.html`.

Own Publications

[233] C. Chen, **H. Wu**, K. Kühnlenz, and S. Hirche, "Switching control for a networked vision-based control system," *at-Automatisierungstechnik*, vol. 59, no. 2, pp. 124–133, 2011.

[234] G. Batz, A. Yaqub, **H. Wu**, K. Kühnlenz, D. Wollherr, and M. Buss, "Dynamic manipulation: Nonprehensile ball catching," in *Proceedings of the 18th Mediterranean Conference on Control & Automation (MED)*, pp. 365–370, 2010.

[235] K. Kühnlenz, **H. Wu**, M. Buss, A. Borst, and T. Zhang, "FPGA design and implementation of insect-inspired Reichardt motion detector and receptive field," *Journal of Image and Graphics (Chin.)*, vol. 14, no. 12, pp. 2489–2496, 2009.

[236] T. Zhang, **H. Wu**, A. Borst, K. Kuhnlenz, and M. Buss, "An FPGA implementation of insect-inspired motion detector for high-speed vision systems," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 335–340, 2008.

[237] L. Zhang, T. Zhang, **H. Wu**, A. Borst, and K. Kühnlenz, "Visual flight control of a quadrotor using bio-inspired motion detector," *International Journal of Navigation and Observation*, vol. 2012, 2012.

[238] **H. Wu**, L. Lou, C.-C. Chen, S. Hirche, and K. Kühnlenz, "Cloud-based networked visual servo control," *IEEE Transactions on Industrial Electronics*, (2012). To appear.

[239] **H. Wu**, K. Zou, T. Zhang, A. Borst, and K. Kühnlenz, "Insect-inspired high-speed motion vision system for robot control," *Biological Cybernetics*, (2012). To appear.

[240] **H. Wu**, L. Lou, C. Chen, S. Hirche, and K. Kuhnlenz, "Performance-oriented networked visual servo control with sending rate scheduling," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6180–6185, 2011.

[241] **H. Wu**, C. Chen, J. Feng, K. Kuhnlenz, and S. Hirche, "A switching control law for a networked visual servo control system," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5556–5563, 2010.

[242] **H. Wu**, L. Lou, C. Chen, K. Kuhnlenz, and S. Hirche, "Distributed computation and data scheduling for networked visual servo control systems," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6216–6221, 2010.

[243] **H. Wu**, L. Lou, C. Chen, S. Hirche, and K. Kuhnlenz, "A framework of networked visual servo control system with distributed computation," in *Proceedings of the 11th International Conference on Control Automation Robotics & Vision (ICARCV)*, pp. 1466–1471, 2010.

[244] **H. Wu**, T. Zhang, A. Borst, K. Kuhnlenz, and M. Buss, "An explorative study of visual servo control with insect-inspired reichardt-model," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 345–350, 2009.

[245] **H. Wu**, T. Zhang, A. Borst, K. Kuhnlenz, and M. Buss, "An explorative study of applying reichardt-model in visual servo control," in *Proceedings of the 1th International Workshop on Cognition for Technical Systems (CoTeSys)*, 2008.