

Adriano A. G. Siqueira · Marco H. Terra  
Marcel Bergerman

# Robust Control of Robots

Fault Tolerant Approaches



Adriano A. G. Siqueira · Marco H. Terra  
Marcel Bergerman

# Robust Control of Robots

Fault Tolerant Approaches



Dr. Adriano A. G. Siqueira  
Engineering School of São Carlos  
University of São Paulo  
Av. Trabalhador Sãocarlense  
400 São Carlos  
SP 13566-590  
Brazil  
e-mail: [siqueira@sc.usp.br](mailto:siqueira@sc.usp.br)

Dr. Marcel Bergerman  
CMU Robotics Institute  
5000 Forbes Avenue  
Pittsburgh  
PA 15213-3890  
USA  
e-mail: [marcel@cmu.edu](mailto:marcel@cmu.edu)

Dr. Marco H. Terra  
Engineering School of São Carlos  
University of São Paulo  
Av. Trabalhador Sãocarlense  
400 São Carlos  
SP 13566-590  
Brazil  
e-mail: [terra@sc.usp.br](mailto:terra@sc.usp.br)

Additional material to this book can be downloaded from <http://extra.springer.com>

ISBN 978-0-85729-897-3  
DOI 10.1007/978-0-85729-898-0  
Springer London Dordrecht Heidelberg New York

e-ISBN 978-0-85729-898-0

British Library Cataloguing in Publication Data  
A catalogue record for this book is available from the British Library

© Springer-Verlag London Limited 2011

MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

Apart from any fair dealing for the purposes of research or private study, or criticism or review, as permitted under the Copyright, Designs and Patents Act 1988, this publication may only be reproduced, stored or transmitted, in any form or by any means, with the prior permission in writing of the publishers, or in the case of reprographic reproduction in accordance with the terms of licenses issued by the Copyright Licensing Agency. Enquiries concerning reproduction outside those terms should be sent to the publishers.

The use of registered names, trademarks, etc., in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant laws and regulations and therefore free for general use.

The publisher makes no representation, express or implied, with regard to the accuracy of the information contained in this book and cannot accept any legal responsibility or liability for any errors or omissions that may be made.

*Cover design:* eStudio Calamar S.L.

Printed on acid-free paper

Springer is part of Springer Science+Business Media ([www.springer.com](http://www.springer.com))

*To My Wife Flaviane and My Son João Vítor*  
A. A. G. S.

*To My Son Diego*  
M. H. T.

*To My Family*  
M. B.

# Preface

The world is teeming with machines that grow crops, process our food, drive us to work, assemble products, clean our homes, and perform thousands of other daily tasks. They are complex systems built on a myriad of electronic, mechanic, and software components, each one prone to malfunction and even fail at any given time. Robust, fault tolerant control of such machines is key to guaranteeing their performance and avoiding accidents. Robotic manipulators, in particular, are especially important when it comes to robust, fault tolerant control. Our society relies on these machines for a large variety of industrial operations; any unscheduled downtime caused by a faulty component can have significant economic costs—not to mention the consequences of a potential injury.

Robust and fault tolerant systems have been studied extensively by academic and industrial researchers and many different design procedures have been developed in order to satisfy rigorous robustness criteria. An important class of robust control methods, introduced by G. Zames in 1980, is based on  $\mathcal{H}_\infty$  theory. The main concept behind this approach is the robustness of the control system to internal uncertainties and exogenous disturbances. Hundreds of works were written extending the seminal results obtained by Prof. Zames. Some of them are sufficiently elegant and effective to be of value in industrial environments. Transforming theory into practice, however, is not a trivial task, as the mathematics involved in robust control can be daunting. This monograph proposes to bridge the gap between robust control theory and applications, with a special focus on robotic manipulators.

The book is organized in nine chapters. In Chap. 1 we present the experimental robot manipulator system used throughout the book to illustrate the various control methodologies discussed. We also present there the simulation and control environment we use to develop and test the methodologies. The environment, named CERob for Control Environment for Robots, is a freeware included with this book and available at <http://extras.springer.com>. The remaining eight chapters are divided in three parts. Part 1 (Chaps. 2–4) deals with robust control of regular, fully-actuated robotic manipulators. Part 2 (Chaps. 5–6) deals with robust fault tolerant control of robotic manipulators, especially the post-failure control

problem. Finally, Part 3 (Chaps. 7–9) deals with robust control of cooperative robotic manipulators.

In Chaps. 2, 3, and 4 we present model-based linear, non-linear, adaptive, and neural network-based  $\mathcal{H}_\infty$  controllers for robotic manipulators. Models based on the Lagrange–Euler formulation and neural networks are used to enable robust control of robots where performance, stability and convergence are guaranteed. One interesting scenario in robot modeling is when the neural network works as a complement of the Lagrange–Euler equations to decrease modeling errors. In these chapters we also explore the use of output feedback controllers, motivated by the fact that in some cases sensors are not available to measure the full array of variables needed for robot control.

In Chaps. 5 and 6 we present strategies to control the position of underactuated manipulators, or manipulators equipped with both regular (active) and failed (passive) joints based on linear parameter-varying models and linear matrix inequalities, and also on game theory. The objective in these chapters is to demonstrate that the system is able to reject disturbances while achieving good position tracking performance. For robotic systems subject to faults, we present a fault tolerant methodology based on linear systems subject to Markovian jumps. We describe in detail the design of  $\mathcal{H}_2$ ,  $\mathcal{H}_\infty$ , and mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  trajectory-following controllers for manipulators subject to several consecutive faults.

In Chaps. 7, 8, and 9 we consider actuated and underactuated cooperative manipulators. One of the most important issues in the robust control of cooperative manipulators is the relationship between disturbance rejection and control of squeeze forces on the load, particularly when the manipulator is underactuated.

Throughout the book we illustrate the concepts presented with simulations and experiments conducted with two 3-link planar robotic manipulators especially designed to pose as fully-actuated or underactuated devices.

São Carlos, Brazil  
 São Carlos, Brazil  
 Pittsburgh, USA

Adriano A. G. Siqueira  
 Marco H. Terra  
 Marcel Bergerman

# Acknowledgments

This book would not have been possible without the help and support of our colleagues and former students from the Mechatronics Laboratory and Intelligent Systems Laboratory at the University of São Paulo. In particular, we would like to express our gratitude to our research partner Prof. Renato Tinós for his contributions to the material presented in Chaps. 7 and 8. We also express our appreciation for the continued support of the Research Council of the State of São Paulo (FAPESP), under grants 98/0649-5, 99/10031-1, 01/12943-0, and the Brazilian National Research Council, under grant 481106/2004-9.

The material presented in this book was partially published previously in scientific journals. We would like to thank the publishers for permitting the reproduction of that content, especially: Cambridge University Press for the paper “Nonlinear  $\mathcal{H}_\infty$  controllers for underactuated cooperative manipulators,” by Adriano A. G. Siqueira and Marco H. Terra, published in *Robotica*, vol. 25, no. 4, in 2007; Elsevier for the papers “Nonlinear mixed control applied to manipulators via actuation redundancy,” by Adriano A. G. Siqueira, Marco H. Terra, and Benedito C. O. Maciel, published in *Control Engineering Practice*, vol. 14, no. 4, in 2006; “A fault tolerance framework for cooperative robotic manipulators,” by Renato Tinós, Marco H. Terra, and Marcel Bergerman, published in *Control Engineering Practice*, vol. 15, no. 5, in 2007; and “Neural network-based control for fully actuated and underactuated cooperative manipulators,” by Adriano A. G. Siqueira and Marco H. Terra, published in *Control Engineering Practice*, vol. 17, no. 3, in 2009; IEEE for the papers “Nonlinear and Markovian controls of underactuated manipulators,” by Adriano A. G. Siqueira and Marco H. Terra, published in *IEEE Transactions on Control Systems Technology*, vol. 12, no. 6, in 2004; “Motion and force control of cooperative robotic manipulators with passive joints,” by Renato Tinós, Marco H. Terra, and João Y. Ishihara, published in *IEEE Transactions on Control Systems Technology*, vol. 14, no. 4, in 2006; and “A fault-tolerant manipulator robot based on  $\mathcal{H}_2$ ,  $\mathcal{H}_\infty$  and mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  Markovian controls,” by Adriano A. G. Siqueira and Marco H. Terra, published in *IEEE/ASME Transactions on Mechatronics*, vol. 14, no. 2, in 2009.

# Acronyms

AAA	Active-active-active
AAP	Active-active-passive
ADP	Adaptive controller
ANN	Artificial neural network
APA	Active-passive-active
APP	Active-passive-passive
CERob	Control environment for robots
CM	Center of mass
CMCE	Cooperative manipulator control environment
DC	Direct current
DLCC	Dynamic load carrying capacity
dll	Dynamically linked library
DOF	Degree of freedom
FDI	Fault detection and isolation
FSJF	Free-swinging joint fault
FTMCE	Fault tolerant manipulator control environment
GTH	Game theory controller
HBC	Hybrid controller
HIN	Linear $\mathcal{H}_\infty$ controller
JPF	Joint position fault
JVF	Joint velocity fault
LJF	Locked joint fault
LMI	Linear matrix inequality
LPV	Linear parameter-varying
MJLS	Markov jump linear systems
MLP	Multilayer perceptron
MTD	Mean time to detection
NET	Neural network controller
NLH	Nonlinear $\mathcal{H}_\infty$ controller
PAA	Passive-active-active
PAP	Passive-active-passive



PCI	Peripheral component interconnect
PD	Proportional-derivative
PPA	Passive-passive-active
RBFN	Radial basis function network
UARM	Underactuated robotic manipulator
UMCE	Underactuated manipulator control environment
VSC	Variable structure controller

# Contents

<b>1</b>	<b>Experimental Set Up</b>	1
1.1	Introduction	1
1.2	UARM Experimental Manipulator	1
1.2.1	Hardware	1
1.2.2	Dynamic Model	5
1.3	Control Environment for Robots (CERob)	6
	References	14

## Part I Fully Actuated Robot Manipulators

<b>2</b>	<b>Linear <math>\mathcal{H}_\infty</math> Control</b>	17
2.1	Introduction	17
2.2	Combined Computed Torque Plus Linear Robust Control	17
2.3	Linear $\mathcal{H}_\infty$ Controllers	22
2.4	Examples	24
2.4.1	Design Procedures	24
2.4.2	Experimental Results	28
	References	33
<b>3</b>	<b>Nonlinear <math>\mathcal{H}_\infty</math> Control</b>	35
3.1	Introduction	35
3.2	Quasi-LPV Representation of Robot Manipulators	36
3.3	$\mathcal{H}_\infty$ Control via Game Theory	37
3.3.1	$\mathcal{H}_\infty$ Control	37
3.3.2	Mixed $\mathcal{H}_2/\mathcal{H}_\infty$ Control	40
3.4	$\mathcal{H}_\infty$ Control via Linear Matrix Inequalities	42
3.4.1	State Feedback $\mathcal{H}_\infty$ Control for LPV Systems	43
3.4.2	Output Feedback $\mathcal{H}_\infty$ LPV Control	45

3.5	Examples	47
3.5.1	Design Procedures	48
3.5.2	Controller Design via Game Theory	53
3.5.3	Controller Design via Linear Matrix Inequalities	55
	References	58
<b>4</b>	<b>Adaptive Nonlinear <math>\mathcal{H}_\infty</math> Control</b>	<b>59</b>
4.1	Introduction	59
4.2	Adaptive $\mathcal{H}_\infty$ Control	60
4.3	Neural Network-Based $\mathcal{H}_\infty$ Control	63
4.4	Examples	65
4.4.1	Design Procedures	66
4.4.2	Model-Based Controller	69
4.4.3	Neural Network-Based Controller	70
	References	73
 <b>Part II Fault Tolerant Control of Robot Manipulators</b>		
<b>5</b>	<b>Underactuated Robot Manipulators</b>	<b>77</b>
5.1	Introduction	77
5.2	Quasi-LPV Representation of Underactuated Manipulators	78
5.3	$\mathcal{H}_\infty$ Control via Linear Matrix Inequalities	80
5.4	$\mathcal{H}_\infty$ Control via Game Theory	82
5.5	Adaptive $\mathcal{H}_\infty$ Control	83
5.5.1	Model-Based Control	84
5.5.2	Neural Network-Based Control	86
5.6	Examples	86
5.6.1	Design Procedure	86
5.6.2	LMI-Based Control	88
5.6.3	Game Theory-Based Control	90
5.6.4	Model-Based Adaptive Controller	93
5.6.5	Neural Network-Based Adaptive Controller	95
	Appendix	97
	References	98
<b>6</b>	<b>Markov Jump Linear Systems-Based Control</b>	<b>101</b>
6.1	Introduction	101
6.2	Motivation	102
6.3	Markovian Jump Linear Model	103
6.3.1	Configuration After Fault Occurrence	105
6.3.2	Linearization Points	106
6.3.3	Markovian States	106
6.3.4	AAA-APA Fault Sequence	107
6.3.5	AAA-PAA-PAP Fault Sequence	110

6.4 MJLS Robust Control Based on State Feedback . . . . . 112

6.4.1  $\mathcal{H}_2$  Control . . . . . 113

6.4.2  $\mathcal{H}_\infty$  Control . . . . . 113

6.4.3 Mixed  $\mathcal{H}_2/\mathcal{H}_\infty$  Control . . . . . 114

6.5 MJLS  $\mathcal{H}_\infty$  Output Feedback Control . . . . . 115

6.6 Examples . . . . . 117

6.6.1 Design Procedures . . . . . 118

6.6.2 AAA–APA Fault Sequence: State-Feedback Control . . . . . 127

6.6.3 AAA–APA Fault Sequence: Output-Feedback Control . . . . . 132

6.6.4 AAA–PAA–PAP Fault Sequence: State-Feedback Control . . . . . 135

6.6.5 AAA–PAA–PAP Fault Sequence: Output-Feedback Control . . . . . 136

Appendix . . . . . 142

References . . . . . 149

**Part III Cooperative Robot Manipulators**

**7 Underactuated Cooperative Manipulators . . . . . 153**

7.1 Introduction . . . . . 153

7.2 Cooperative Manipulators . . . . . 154

7.3 Control of Cooperative Arms with Passive Joints . . . . . 156

7.3.1 Motion Control . . . . . 158

7.3.2 Squeeze Force Control . . . . . 161

7.4 Dynamic Load-Carrying Capacity of Cooperative Manipulators with Passive Joints . . . . . 163

7.5 Examples . . . . . 165

7.5.1 Design Procedures . . . . . 165

7.5.2 Motion and Squeeze Force Control . . . . . 174

7.5.3 DLCC of Two 3-Joint Manipulators with Passive Joints . . . . . 175

References . . . . . 176

**8 A Fault Tolerance Framework for Cooperative Robotic Manipulators . . . . . 177**

8.1 Introduction . . . . . 177

8.2 Cooperative Manipulators . . . . . 178

8.3 Fault Tolerance System . . . . . 179

8.4 Fault Detection and Isolation System . . . . . 180

8.4.1 Incorrectly-Measured Joint Position and Joint Velocity Faults . . . . . 180

8.4.2 Free-Swinging and Locked Joint Faults . . . . . 182

8.5	Control Reconfiguration . . . . .	184
8.5.1	Incorrectly-Measured Joint Velocity and Position Faults . . . . .	184
8.5.2	Free-Swinging Joint Faults . . . . .	185
8.5.3	Locked Joint Faults . . . . .	186
8.6	Examples . . . . .	187
8.6.1	Design Procedures . . . . .	187
8.6.2	FDI Results . . . . .	190
	References . . . . .	194
<b>9</b>	<b>Robust Control of Cooperative Manipulators . . . . .</b>	<b>197</b>
9.1	Introduction . . . . .	197
9.2	Fully-Actuated Cooperative Manipulators . . . . .	198
9.3	Underactuated Cooperative Manipulators . . . . .	200
9.4	Motion and Squeeze Force Control . . . . .	202
9.5	Nonlinear $\mathcal{H}_\infty$ Control . . . . .	203
9.5.1	Control Design via Quasi-LPV Representation . . . . .	203
9.5.2	Control Design via Game Theory . . . . .	204
9.6	Neural Network-Based Adaptive Nonlinear $\mathcal{H}_\infty$ Control . . . . .	205
9.7	Examples . . . . .	211
9.7.1	Design Procedures . . . . .	211
9.7.2	Fully-Actuated Configuration . . . . .	214
9.7.3	Underactuated Configuration . . . . .	219
9.8	Neural Network-Based Adaptive Controller . . . . .	219
	References . . . . .	225
	<b>Index . . . . .</b>	<b>227</b>

# Chapter 1

## Experimental Set Up

### 1.1 Introduction

The field of Robotics is, by its very nature, an experimental one. No robot control methodology can be deemed to perform satisfactorily if it has not been validated on an actual physical system. In this book we illustrate all control methods presented by applying them to custom-designed robotic manipulators.

The manipulators are two 3-link open-chain, serial link arms built by Ben Brown, Jr. from Pittsburgh, PA, USA, which we name the UARMs, or UnderActuated Robotic Manipulators (Fig. 1.1). The two most salient features of these manipulators are that their joint motors possess very low friction and are equipped with on/off brakes, thus allowing us to simulate a variety of joint failure conditions. We created an open source MATLAB<sup>®</sup>-based UARM simulator that readers can utilize to validate the control methodologies presented throughout the chapters. The simulator includes also the Matlab source code for all methods described. This Control Environment for Robots (CERob) is in fact more than just a standard simulator; in our laboratory, control methodologies can be validated in the virtual manipulators and then transferred to the actual ones at the click of a button.

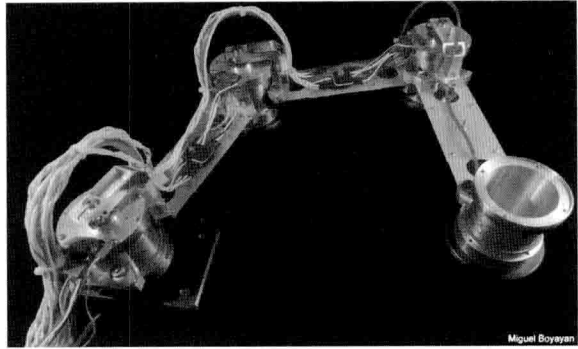
In the first part of this chapter we describe in detail the UARM hardware and its dynamic model. In the second part we describe the basic functionality of CERob. Specific details on CERob as it applies to particular controllers are presented in the pertinent chapters.

### 1.2 UARM Experimental Manipulator

#### 1.2.1 Hardware

Each UARM is a 3-link planar, open-chain, serial-link manipulator. They are equipped with low-friction DC motors directly connected to the links, with no gearboxes. When the motors are powered, the joints behave as regular

**Fig. 1.1** UnderActuated  
Robot Manipulator (UARM)



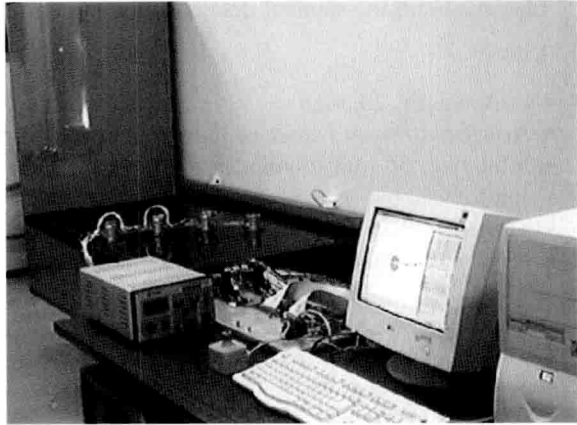
fully-actuated ones; and when they are not powered, the joints can move essentially freely and dub as unactuated or passive joints. The manipulator's configuration (fully-actuated or underactuated) can then be defined at will in real time by simply powering or not each joint. Additionally, the joints are equipped with diaphragm-based on/off brakes, which can be used to simulate locking-type faults or to enable underactuated manipulator position control. Joints are numbered from 1 to 3, with joint 1 fixed to a smooth marble table. A dummy load can be attached to the end-effector's cup-shaped housing to provide for meaningful manipulation experiments. The entire system resides on a horizontal plane and runs on a thin air film that reduces to practically zero the friction with the table. Pressurized air pumps and computer-controlled solenoid valves complete the hardware of the system.

Incremental encoders with quadrature decoding, located at the top of each joint, are used to measure the relative angular joint positions. The angular velocities are computed via numerical differentiation and low-pass filtering. Such procedure is known to result in measurement noise and can therefore lead to poor position control performance. This is one of our motivations to use output feedback control laws, where only joint position measurements are used.

An equipment board provides a mounting surface for the motor amplifiers, valves and air pressure regulators for brake and flotation air, and a custom interface board. Two feeding voltages are supplied by a power unit: 48 V/20 A for the motors, and 24 V/1 A for the interface board. A kill switch ("emergency stop") mounted in a small yellow enclosure can be located remotely, and controls the power supply to the interface board and amplifiers. The equipment board also provides a location where the UARM can be secured for safe transport without disconnecting it from the rest of the system. Figure 1.2 shows the complete system, with the robot manipulator, the equipment board, the power supply unit, and the control computer.

The communication between the control computer and the hardware is performed by a PCI (Peripheral Component Interconnect) input-output board from Motenc. This device is able to control up to eight servo motors simultaneously

**Fig. 1.2** Complete UARM hardware setup



(eight robot joints). The Motenc board connects to the custom interface board via two 50-conductor ribbon cables. The features available on this board include:

- 8 differential encoder inputs, 32-bit resolution;
- 8 analog outputs,  $\pm 10$  V range, 13-bit resolution;
- 8 analog inputs,  $\pm 5$  V range, 14-bit resolution;
- 100 digital I/O (68 inputs and 32 outputs) in four 50-pin headers, opto-22 compatible;
- +5 V available on headers, fused (resetable), max current 2 A;
- Programmable timer interrupts;
- Watchdog timer;
- Hardware board ID for multiple board applications;
- Filters at digital inputs to remove high frequency noise; and
- Hardware ESTOPs.

The designer can access the board I/O channels by using the *dll* library provided by the manufacturer. Table 1.1 shows the MATLAB<sup>®</sup>-based functions we developed to communicate with the UARM hardware. Most of the time only four commands are necessary to control the robot: those to set the desired voltages, read the current encoders' values, reset the encoders, and activate or release the brakes, at the same time activating or inhibiting the motors.

**Table 1.1** MATLAB<sup>®</sup>-based functions for control of the UARM

Function	Description
<code>set_dac_all_stg([v1 v2 v3])</code>	Set the desired voltages to the DC motors
<code>get_position</code>	Read joint encoders
<code>set_encoder_one_stg(enc, value)</code>	Set encoder to the specified value
<code>setbrakemotor(value)</code>	Activate/release brakes and activate/inhibit motors



Electrical and mechanical details of the system are as follows:

#### 1. Links

- Link length: 20.3 cm
- Arm length from center of joint 1 to center of tip: 60.96 cm
- Joint size: 76 mm (diameter)  $\times$  86 mm (height)
- Joint mass: 670 g
- Tip mass: 220 g (default, customized by user)
- Link mass: 30 g (excluding wires, air hoses, and connectors).

#### 2. Joint motors

- Model: Kollmorgen RBE-1213 brushless DC
- Nominal voltage: 48 V CC
- Winding resistance:  $2.4 \Omega$
- Torque constant: 0.14 Nm/A
- Peak torque: 2.8 Nm
- Back EMF constant: 15 V/kRPM
- Continuous stall torque: 0.35 Nm
- Motor mass: 344 g
- Rotor inertia:  $0.0000148 \text{ kg m}^2$ .

#### 3. Motor amplifiers

- Model: Elmo SBA 10/100H-4
- Peak current: 20 A
- Continuous current: 10 A
- Supply voltage: 20–90 V CC
- Current-to-voltage constant (adjustable): 1.61 A/V.

#### 4. Brakes

- Type: air actuated diaphragm
- Pressure: 100 psi (700 kPa) max.
- Valves: Clippard model EVO-3M, 24 VDC, 0.67 W
- Torque: 2.8 Nm max.

#### 5. Optical encoders

- Model: Hewlett Packard HEDS 9040-T00
- Disk: HEDS 6140-T08
- Lines: 2000/revolution
- Counts: 8000/revolution after quadrature decoding.

#### 6. Air bearings

- Orifice diameter: 0.36 mm
- Air gap: approx. 0.08 mm
- Air supply pressure: 100 psi (700 kPa) max.