

Contents

Acknowledgments	i
Abstract	ii
Résumé	iii
Introduction	1
I Background	3
1 Theoretical Background	4
1.1 Definitions and Terminology	4
1.2 Motion and Task Planning	5
1.3 Manipulator Control	6
1.3.1 Position Control vs. Torque Control	7
1.3.2 Advantages of Torque Control	7
1.3.3 Gravity Compensation	7
1.3.4 Friction	8
1.4 Practical Implementation Considerations	10
2 State-of-the-Art on Peg-in-Hole assembly	13
2.1 Assembly process	13
2.2 Geometric classification	14
2.3 Control methods	15
2.3.1 Intuitive assembly strategies	16
2.3.2 Compliance methods	17
2.3.3 Auxiliary technologies	19
2.4 Summary and Method Selection	20
3 Introduction to the Virtual Mechanism Control Framework	21
3.1 Virtual mechanisms fundamentals	21
3.1.1 Virtual forces	21
3.1.2 Virtual elements	22
3.1.3 Virtual Structure and Implementation	24
3.2 Example of design of Virtual mechanism	25
3.2.1 Virtual model control for reaching	25
3.2.2 Virtual model control for grasping	26
3.3 A Word on Passivity	26
3.3.1 Analogy with an RLC Circuit	26
3.3.2 Consequence for Stability	27

II	Application	28
4	Experimental Framework: Platform, Interface, and Protocol	29
4.1	Robotic Platform and Designed Components	29
4.2	Software Architecture	30
4.2.1	Control Algorithm : Julia and VMRobotControl.jl	30
4.2.2	Middleware : ROS2	30
4.2.3	Translator : the ur_driver	31
4.3	Problem Formulation	31
4.3.1	Scope and Assumptions	31
4.3.2	Evaluation Criteria	32
4.4	Experimental Protocol	32
5	Virtual Mechanism Control for Peg-in-Hole Assembly: Design and Experimental Validation	34
5.1	Controller Design	34
5.1.1	Task Decomposition and Trajectory Planning	34
5.1.2	Virtual Mechanism Design	36
5.2	Simulation	38
5.2.1	Choice of Simulator	38
5.2.2	Robot Model	38
5.2.3	Results	39
5.3	Experimental Results	43
5.3.1	Influence of Stiffness on Positioning Accuracy	43
5.3.2	The Sim-to-Real Gap and the Role of Friction	44
5.3.3	Reaching on the Real Robot	44
5.3.4	Peg-in-Hole	46
5.3.5	Discussion	53
III	Conclusions and perspectives	55
6	Conclusions	56
7	Perspectives	58
7.1	Oriented toward Cilyx	58
7.1.1	Extension to Position-Controlled Robots	58
7.1.2	Pushing the Approach to the Next Level	58
7.2	Oriented toward the Neuroengineering Lab of ULIEGE	59
7.2.1	Data-Driven Friction Compensation	59
7.2.2	Porting the VMC Toolbox to Accessible Languages	59
IV	APPENDICES	60
8	Appendices	61
Appendices		61
8.1	Software Interface	61
8.1.1	Overview of the Software Stack	61
8.1.2	Step 1: Software Requirements	61
8.1.3	Step 2: Network Setup	62
8.1.4	Step 3: Robot Calibration	62

8.1.5	Step 4: Installing the UR Robot Driver	62
8.1.6	Step 5: Launching the Driver	62
8.1.7	Step 6: Switching to Torque Control	63
8.1.8	Step 7: Sending Torque Commands	63
8.1.9	Step 8: Reading the Robot State	64
8.1.10	Step 9: Launching the Julia Controller	64
8.1.11	Quick Reference	64
8.2	Visual representation	65
Declaration on the Use of Automatic Tools		66
Bibliography		73