

ÖREBRO UNIVERSITY

SCHOOL OF SCIENCE AND TECHNOLOGY, AASS



---

# Control of The Velvet Fingers

– INTERNSHIP FINAL REPORT –

---

*Author:*  
KWIECIŃSKI Krzysztof

*Supervisor:*  
Dr. Robert KRUG

September 23, 2015

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Analysis</b>	<b>1</b>
2.1	DC Armature Model . . . . .	1
<b>3</b>	<b>Implementation</b>	<b>2</b>
3.1	Matlab . . . . .	2
3.2	Arduino . . . . .	2
3.3	Documentation . . . . .	4
<b>4</b>	<b>Case studies</b>	<b>4</b>
4.1	Basic methods . . . . .	4
4.1.1	Current Control . . . . .	4
4.1.2	Position Control . . . . .	5
4.1.3	Velocity Control . . . . .	5
4.2	Complex methods . . . . .	5
4.2.1	Stiffness Control . . . . .	5
4.2.2	Impedance Control . . . . .	7
4.2.3	Parallel Force/Position Control . . . . .	8
<b>5</b>	<b>Conclusion</b>	<b>9</b>

## Abstract

In this paper different approaches to control of a gripper are presented. The main focus is on the current (force) control. A gripper used for experiments is The Velvet Fingers, a simple tactile under-actuated gripper.



Figure 1: The Velvet Fingers

# 1 Introduction

The report treats of effects of my internship that took place at *School of Science and Technology* at the *Örebro University* in Sweden. I worked for *Centre for Applied Autonomous Sensor Systems*, at the *Mobile Robotics and Olfaction Lab*. A time span I spent here was two months, namely August and September 2015. My main goal was to test different ways of control **The Velvet Fingers**. Since the gripper is a smart gripper using a novel concept of the end-effector combining simple under-actuated mechanics with high manipulation possibilities, there is only need to control DC motors.

## 2 Analysis

### 2.1 DC Armature Model

A DC motor can be described by two equations:

$$V(t) = L \frac{di(t)}{dt} + Ri(t) + E(t) \quad (1)$$

$$J\dot{\omega}(t) = k_{tau}i(t) - \mu\omega(t) - T_L(t), \quad (2)$$

where 1 describes electrical and 2 describes mechanical aspect of the motor.

It is known that

$$E(t) = k_{emf}\omega(t), \quad (3)$$

$$T(t) = k_{tau}i(t). \quad (4)$$

Combining the above equations and assuming that changes in current are instantaneous yields

$$V(t) = \frac{R}{k_{tau}}(J\dot{\omega}(t) + \mu\omega(t) + T_L(t)) + k_{emf}\omega(t) \quad (5)$$

or simply

$$V(t) = Ri(t) + k_{emf}\omega(t) \quad (6)$$

if the current is measured.

Equations 5 and 6 hold for every state of the motor. However, since we want the gripper to hold an object firmly, desired  $\omega$  and  $\dot{\omega}$  are zero. That means the final feedforward equation when holding an object can be reduced to

$$V(t) = R \frac{T_L(t)}{k_{tau}} = Ri(t). \quad (7)$$

Nomenclature:  $R$  - motor terminal resistance,  $L$  - motor terminal inductance,  $i$  - current,  $E$  - back electromotive force,  $J$  - rotor inertia moment,  $\omega$  - rotor angular velocity,  $\dot{\omega}$  - rotor angular acceleration,  $k_{tau}$  - torque constant,  $k_{emf}$  - speed constant,  $T_L$  - load torque.

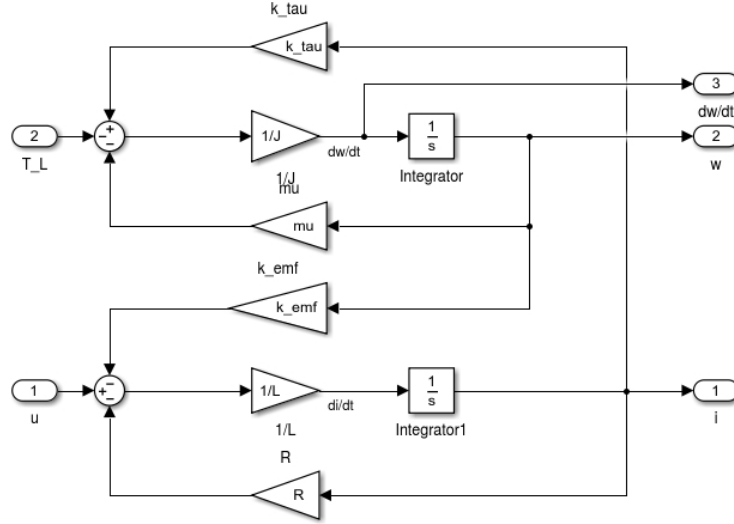


Figure 2: A model of a DC Motor in Simulink built of basic blocks

## 3 Implementation

### 3.1 Matlab

The first step was to understand the working principle of a DC motor. After getting such knowledge, I had to verify it in Matlab and Simulink. I created two models of a DC motor in Simulink: one built of only basic blocks and second using blocks from the Simscape library. Models are shown on figures 2 and 3.

I tested both models using scripts written in Matlab. Inputs of the system were voltage and applied force, and outputs were current, angular velocity and acceleration. Tests were done for open and closed loop. Control of the process variable was done with a PID controller. Results for both models were almost identical and the simulations confirmed theoretical knowledge and developed my intuition.

### 3.2 Arduino

Afer getting some experience, I started working with hardware. The computations and control were done by Arduino Due board. The code was written in C++. With such combination, I was able to write a program containing a lot of operations using a high-level programming language.

Control of speed and direction of the motor's movement was done with use of an external H-Bridge and a PWM signal from Arduino.

Arduino was connected with a computer through a serial port. As a middleware The Robot Operating System was used. ROS allowed to change parameters on-line and provided opportunity to observe and visualize values of chosen variables.

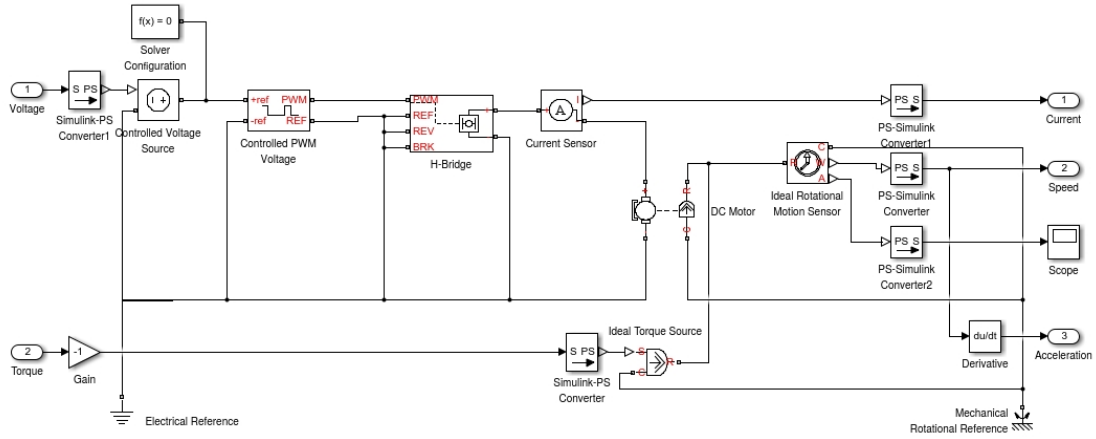


Figure 3: A model of a DC Motor in Simulink built of blocks from the Simscape library

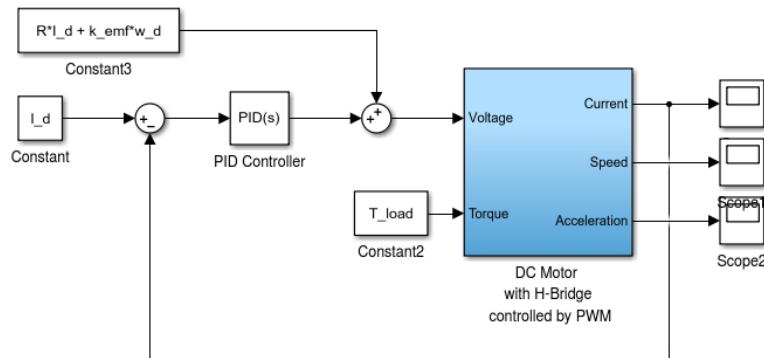


Figure 4: An example control loop with feedforward term

Thanks to usage of encoders and measuring the current flowing through the motor, I was able to implement various types of control, for instance:

- current (force) control,
- position control,
- velocity control,
- stiffness control,
- impedance control,
- parallel force/position control.

Because the control was dedicated for a gripper, the most important was the first one.

### 3.3 Documentation

All files with a detailed documentation of the code generated by Doxygen are available at the GitHub repository.

## 4 Case studies

For conducting experiments two setups were used. A basic one consisted only from one DC motor. The second was The Velvet Fingers.

### 4.1 Basic methods

Three fundamental methods of control are: current, position and velocity control. They all rely on measuring one process variable (PV) and try to reach the setpoint (SP) with use of a PID controller, which output is the control variable (CV).

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de}{dt} \quad (8)$$

#### 4.1.1 Current Control

Current control can be identified with force control. Even though there are no tactile sensors, hence the force is not explicitly measured, the applied force can be implicitly calculated from 4.

Figure 5 presents changes of current flowing through the motor in time. X-axis is time [s], Y-axis is current [A]. We can see three phases: in the beginning and the end the motor was running without any constraints, in the middle phase the movement was constrained. The response was almost instantaneous, and the control variable reached the setpoint very precisely and without oscillations.

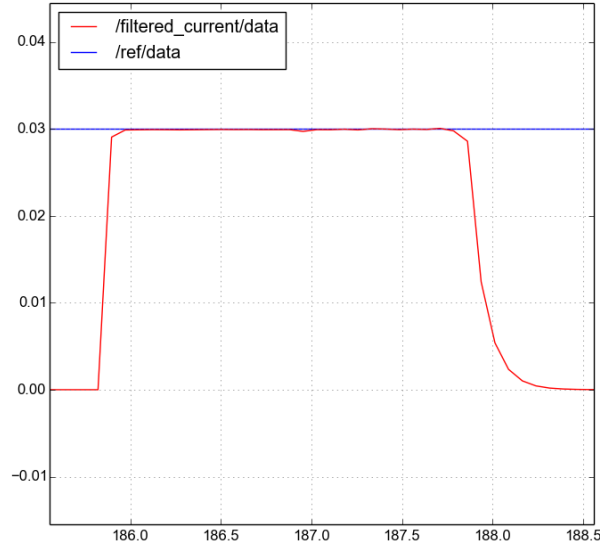


Figure 5: An exaple plot of the current control

#### 4.1.2 Position Control

Figure 6 presents changes of the motor's position in time. X-axis is time [s], Y-axis is position [rad]. After applying the voltage the movement was steady. The control variable reached the setpoint very precise and without oscillations.

#### 4.1.3 Velocity Control

Figure 7 presents changes of the motor's velocity in time. X-axis is time [s], Y-axis is angular velocity [rad/s]. After applying the voltage the motor reached the setpoint very fast. Although oscillations are presented in the plot, the movement of the motor was actually smooth. The oscillations are visible due to measurements done by encoder and derivating the position, which introduce some noise.

### 4.2 Complex methods

When interacting with the environment, there is need for more sophisticated methods. The control variable is calculated based on two or more process variables.

Nomenclature:  $p$  - current position,  $\dot{p}$  - current velocity,  $\ddot{p}$  - current acceleration,  $f$  - current force,  $d$  - desired value,  $K_p$  - position coefficient,  $K_v$  - velocity coefficient,  $K_f$  - force coefficient,  $K_i$  - integral coefficient,  $M$  - mass,  $B$  - damping,  $K$  - stiffness.

#### 4.2.1 Stiffness Control

Stiffness control derives from a position control scheme of PD type.



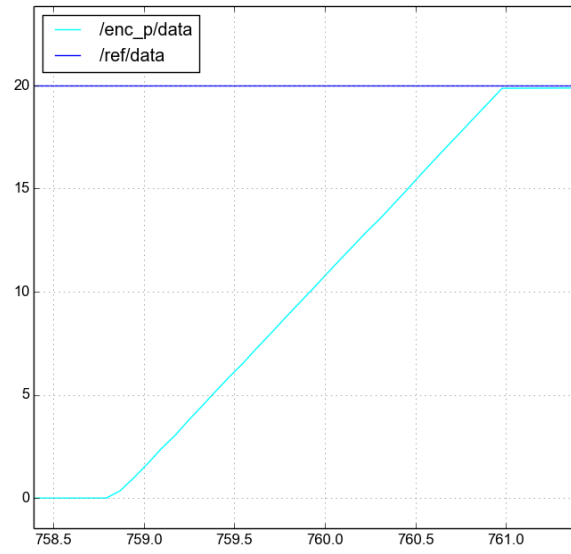


Figure 6: An exaple plot of the position control

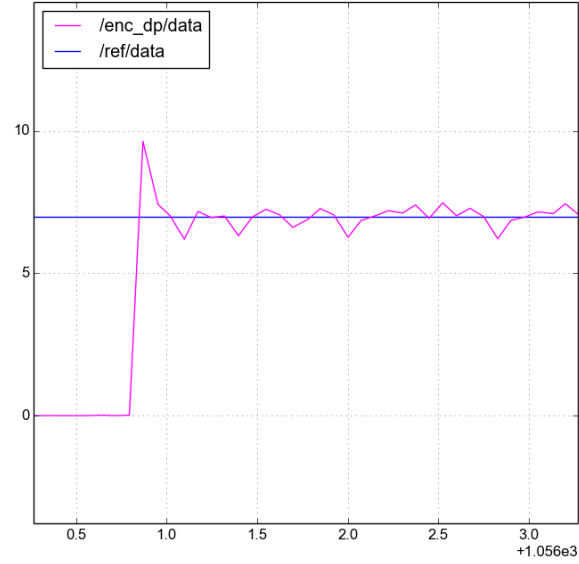


Figure 7: An exaple plot of the velocity control

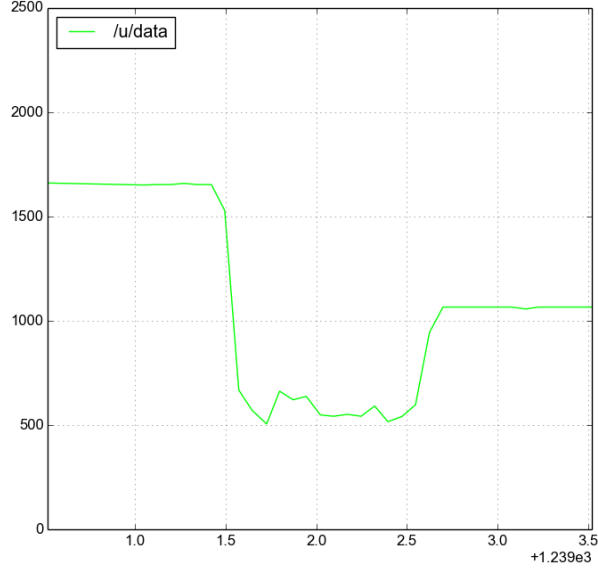


Figure 8: An exaple plot of the stiffness control

$$u(t) = K_p(p_d - p) - K_v\dot{p}, \quad (9)$$

Figure 8 presents changes of the control variable in time. X-axis is time [s], Y-axis is the CV. We can see three phases: in the beginning and the end the movement is constrained, in the middle phase the motor is running without any constraints. When in contact with the environment, the CV is steady and larger than during the unconstrained motion. Since stiffness control derives from a position control, the closer to the setpoint, the lower the CV.

#### 4.2.2 Impedance Control

In impedance control the input is flow (velocity or displacement) and the output is effort (force). In directions where the robot has to be environment-sensitive, impedance is low, otherwise impedance is high. The desired impedance behavior is a trade-off between trajectory error and force error. If the motion is unconstrained, force will be zero and the robot will move to the reference position.

$$u(t) = M(\ddot{p}_d - \ddot{p}) + B(\dot{p}_d - \dot{p}) + K(p_d - p) - K_f f \quad (10)$$

Desired acceleration and velocity are usually not present to guarantee passivity when in contact with the environment.

Figure 9 presents changes of the control variable in time. X-axis is time [s], Y-axis is the CV. We can see three phases: in the beginning and the end the movement is constrained, in the middle phase the motor is running without any constraints. When

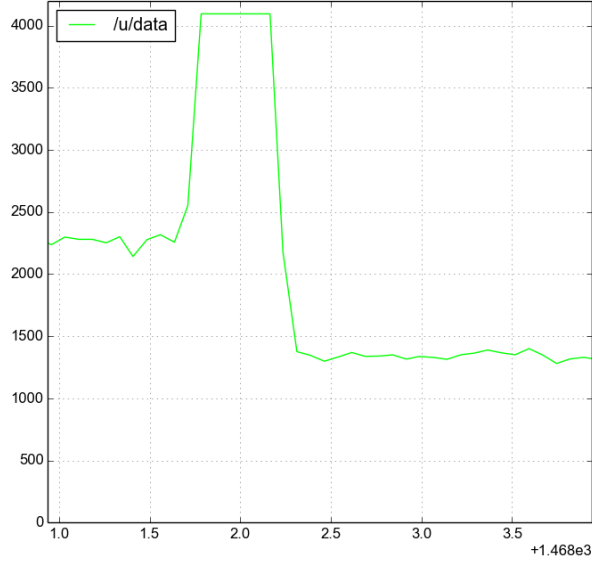


Figure 9: An exaple plot of the impedance control

in contact with the environment, the CV is steady and smaller than during the unconstrained motion. Such behaviour keeps the interaction force in desired boundaries and guarantees reaching the setpoint, if possible.

#### 4.2.3 Parallel Force/Position Control

In order to combine features of stiffness control and force control, a parallel force/position regulator can be designed where a PI force control action plus desired force feedforward is used in parallel to a PD position control action. The position controller is an impedance while the force controller results in a filtering action on the force variables.

$$u(t) = M\ddot{e}_p + K_v\dot{e}_p + K_p e_p + K_f e_f + K_i \int_0^t e_f d\tau, \quad (11)$$

where error  $e$

$$e_{variable}(t) = variable_{desired} - variable(t). \quad (12)$$

Figure 10 presents changes of the control variable in time. X-axis is time [s], Y-axis is the CV. We can see three phases: in the beginning and the end the movement is constrained, in the middle phase the motor is running without any constraints. When in contact with the environment, the CV is steady and smaller than during the unconstrained motion. When moving freely and going in direction of the setpoint, the CV slowly decreases. Such behaviour keeps the interaction force at desired level independently of the position error and guarantees reaching the setpoint, if possible.

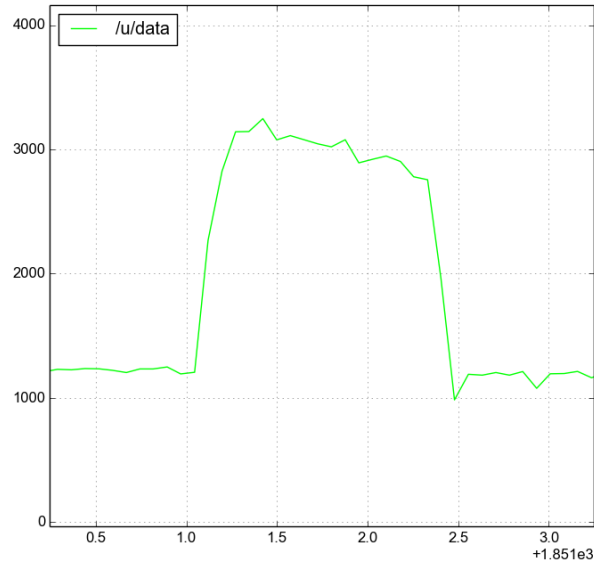


Figure 10: An exaple plot of the parallel force/position control

## 5 Conclusion

There are plenty of methods to control a DC motor. In this report a couple of them were treated.

Working with the environment requires interaction so taking into consideration more than one variable is necessary. The most obvious conclusion is that the more complicated type of control, the more effort is needed to achieve the goal. It is why all the basic methods always worked fast and reliable, and the more complex did not. The problem is achieving two different goals which is practically impossible. All things considered make more complex methods more difficult to implement, however, if properly done, they are the best choice. The point is to find a fair trade-off between goals that should be achieved.

Furthermore, working with hardware might be a source of difficult to track down problems. Due to working with not impeccable measuring equipment, filtering variables, i.e. current and position, is really important. A PID controller plays a crucial role in control. If properly tuned, the controller is fast and precise.

Future research efforts will be devoted to improving all types of control.

## References

- [1] S. Chiaverini and L. Sciavicco. The parallel approach to force/position control of robotic manipulators. *Robotics and Automation, IEEE Transactions on*, 9(4):361–373, Aug 1993.
- [2] S. Chiaverini, B. Siciliano, and L. Villani. A survey of robot interaction control schemes with experimental comparison. *Mechatronics, IEEE/ASME Transactions on*, 4(3):273–285, Sep 1999.
- [3] N. Hogan. Impedance control: An approach to manipulation. In *American Control Conference, 1984*, pages 304–313, June 1984.
- [4] N. Hogan and S. P. Buerger. Impedance and interaction control, 2005.
- [5] R. Krug, T. Stoyanov, M. Bonilla, V. Tincani, N. Vaskevicius, G. Fantoni, A. Birk, A. Lilienthal, and A. Bicchi. Velvet fingers: Grasp planning and execution for an underactuated gripper with active surfaces. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 3669–3675, May 2014.
- [6] V. Tincani, M. Catalano, E. Farnioli, M. Garabini, G. Grioli, G. Fantoni, and A. Bicchi. Velvet fingers: A dexterous gripper with active surfaces. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pages 1257–1263, Oct 2012.
- [7] V. Tincani, M. Catalano, G. Grioli, T. Stoyanov, R. Krug, A. Lilienthal, G. Fantoni, and A. Bicchi. Sensitive active surfaces on the velvet ii dexterous gripper. In *IEEE Int. Conf. on Robotics and Automation (ICRA) - Workshop "Get in touch!" Tactile & force sensing for autonomous, compliant, intelligent robots*, 2015.
- [8] V. Tincani, G. Grioli, M. Catalano, M. Garabini, S. Grechi, G. Fantoni, and A. Bicchi. Implementation and control of the velvet fingers: A dexterous gripper with active surfaces. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2744–2750, May 2013.