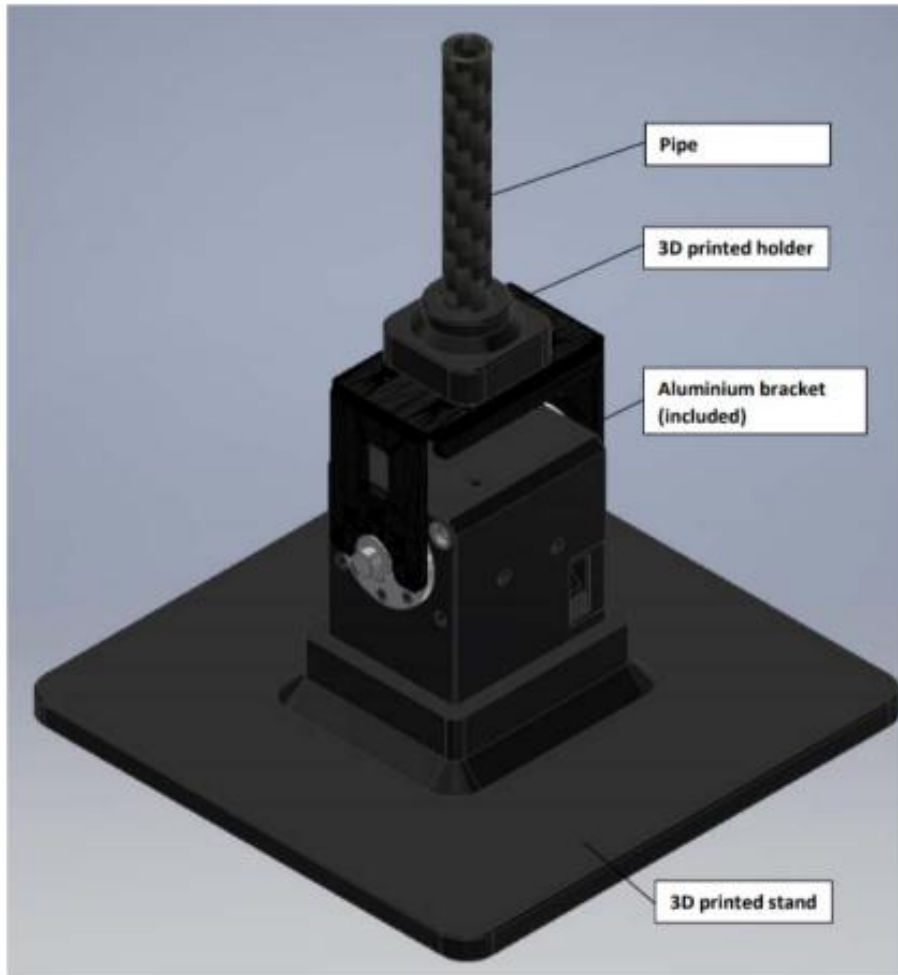


# **Bio-inspired Autonomous Systems**

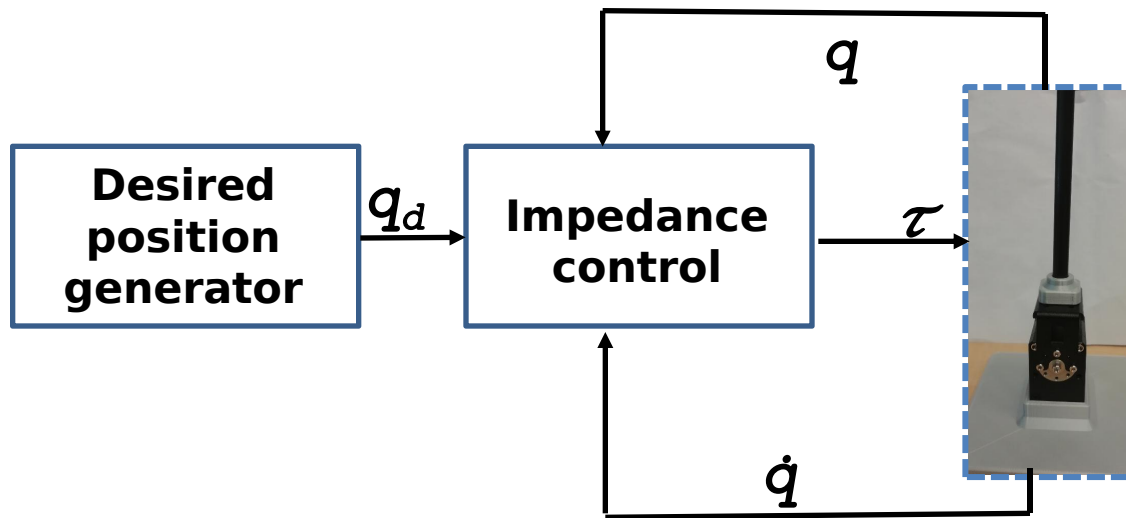
**- Online impedance adaptation control**

Xiaofeng Xiong

# Muscle-like control



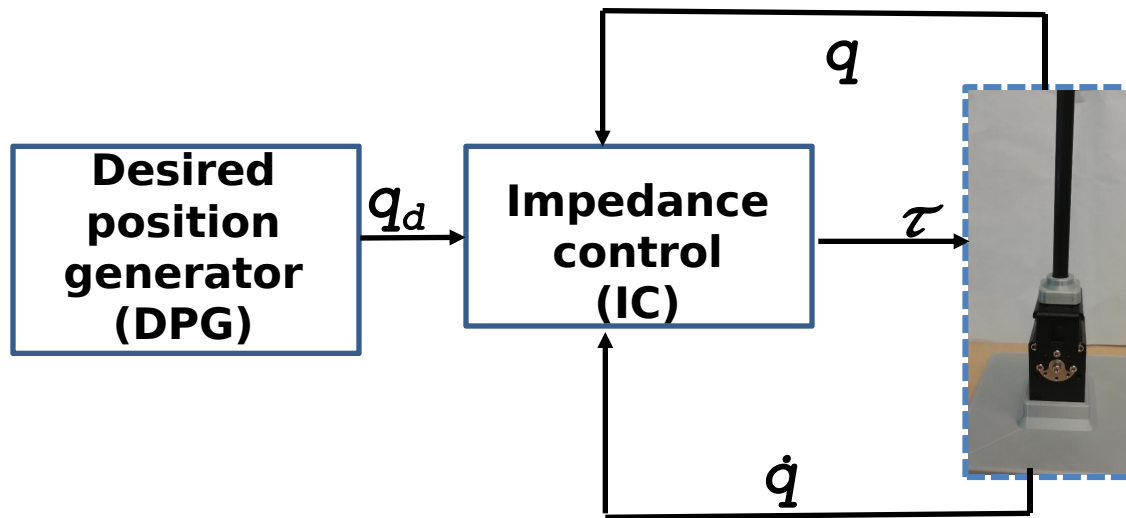
# Muscle-like control



**XM430-W350-R**

<http://emmanual.robotis.com/docs/en/dxl/x/xm430-w350/>

# Muscle-like control



**Two DPGs: posture control and periodic tracking**

**Two ICs: constant control and variable impedance control**

# Muscle-like control

**main.py**

```
'''
Tasks:

T1: see pla_postu_con()

T2: see pla_tra()

T3: see get_pos_diff()

T4: see get_vel_diff()

T5: see get_tra_diff()

T6: see get_coe()

T7 and T8: see const_impe()

T9 and T10: see ada_impe()

'''
```

**Two DPGs: posture control and periodic tracking**

**Two ICs: constant control and variable impedance control**

# Muscle-like control

main.py

```
def pla_postu_con(self, t, ini_pos, tar_pos):# posture control
    #keep initial motor angle without respect to t, ini_pos, tar_pos

    #T1
```

Posture control:  $q_d = 0.0$

# Muscle-like control

main.py

```
def pla_tra(self, t, ini_pos, tar_pos): # periodic tracking
    #generate the periodic function sin with respect to t, ini_pos, tar_pos
    '''
    ini_pos: initial value
    tar_pos: maximum value
    self.cir_fre: frequency
    '''
    #T2
```

**Periodic tracking:  $\sin (ft)$**

# Muscle-like control

main.py

```
def get_pos_diff(self):# angle difference, see Eq.(3)
    for i in range(self.dm.moto_num):
        '''
        self.dm.init_var[i]: initial motor position (i.e., 0 ~ 4095)
        self.dm.now_pos[i]: current motor position (i.e., 0 ~ 4095)
        self.dm.pos2rad[i]: ratio between radius and position

        self.pos_rad[i]: initial motor angle (i.e., -pi ~ pi)
        self.pos_diff[i]: angle difference
        '''
    #T3
```

X. Xiong and P. Manoonpong, "Adaptive Motor Control for Human-like Spatial-temporal Adaptation," 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 2018, pp. 2107-2112.doi: 10.1109/ROBIO.2018.8665222



# Muscle-like control

main.py

```
def get_vel_diff(self):# velocity difference, see Eq.(3)
    for i in range(self.dm.moto_num):
        '''
        self.dm.now_vel[i]: velocity feedback (unit: rad/s)
        self.vel_des[i]: desired velocity

        self.vel_diff[i]: velocity difference
        '''

    #T4
```

# Muscle-like control

main.py

```
def get_tra_diff(self):# tracking difference, see Eq.(3)
    for i in range(self.dm.moto_num):
        '''
        self.pos_diff[i]: angle difference
        self.vel_diff[i]: velocity difference

        self.tra_diff[i]: tracking error
        '''

    #T5
```

# Muscle-like control

main.py

```
def get_coe(self):# adaptation scalar, see Eq.(9)
    for i in range(self.dm.moto_num):
        '''
        self.a and self.b: constants
        self.tra_diff[i]: tracking error

        self.co_diff[i]: adaptation scalar
        '''

    #T6
```

X. Xiong and P. Manoonpong, "Adaptive Motor Control for Human-like Spatial-temporal Adaptation," 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 2018, pp. 2107-2112.doi: 10.1109/ROBIO.2018.8665222

# Muscle-like control

main.py

```
def const_impe(self):# constant impedance control
    #need to update self.pos_diff[i] and self.vel_diff[i]

    #T7

    for i in range(self.dm.moto_num):
        '''
        self.cons_k: stiffness constant
        self.cons_d: damping constant
        self.k[i]: motor control stiffness parameter
        self.d[i]: motor control damping parameter
        self.ff[i]: control bias
        self.tau[i]: control output to the motor
        '''

    #T8
```

X. Xiong and P. Manoonpong, "Adaptive Motor Control for Human-like Spatial-temporal Adaptation," 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 2018, pp. 2107-2112.doi: 10.1109/ROBIO.2018.8665222

# Muscle-like control

main.py

```
def ada_impe(self):# adaptive impedance control
    #need to update self.pos_diff[i], self.vel_diff[i]

    #T9

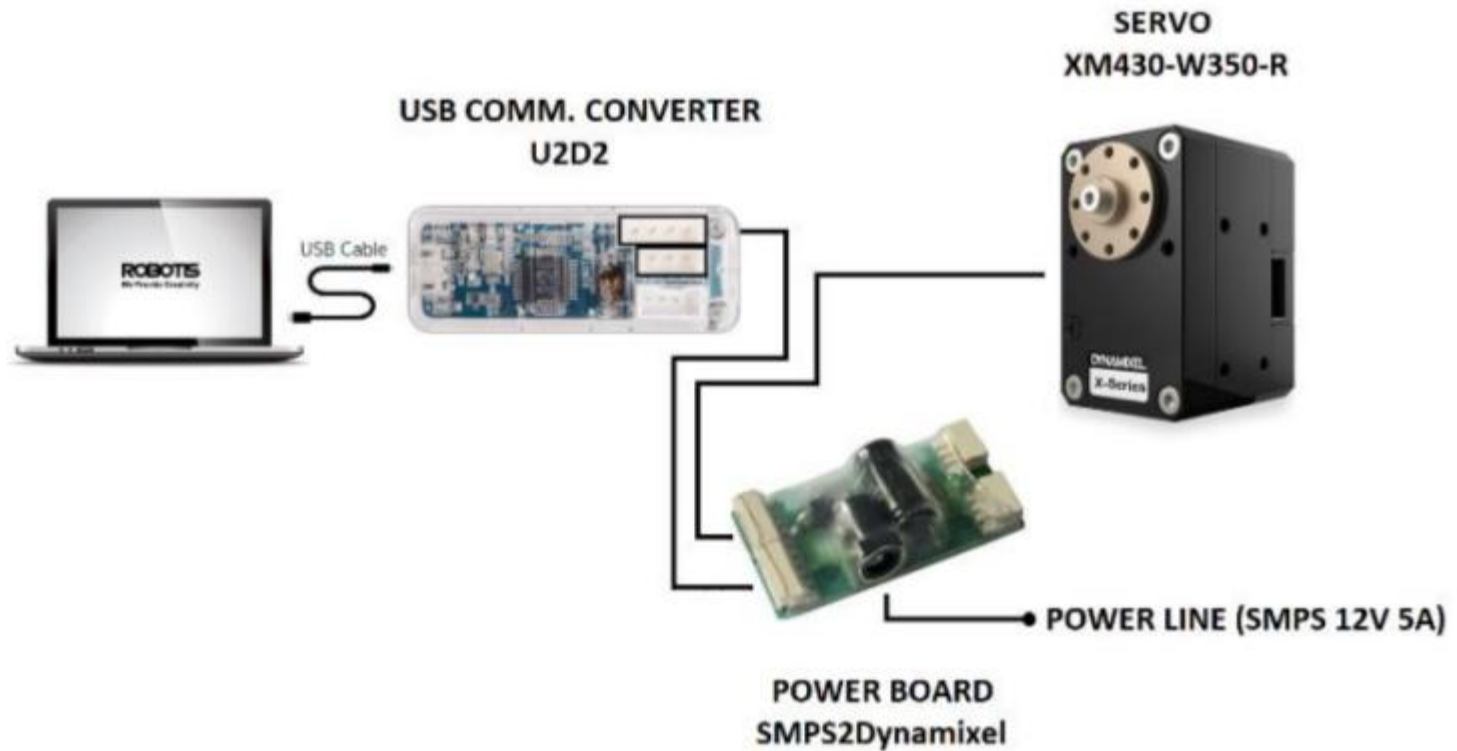
    for i in range(self.dm.moto_num):
        '''
        self.k[i]: motor control stiffness parameter
        self.d[i]: motor control damping parameter
        self.ff[i]: control bias
        self.tau[i]: control output to the motor
        '''

        #need online modulate impedance parameters se

    #T10
```

X. Xiong and P. Manoonpong, "Adaptive Motor Control for Human-like Spatial-temporal Adaptation," 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 2018, pp. 2107-2112.doi: 10.1109/ROBIO.2018.8665222

# Muscle-like control

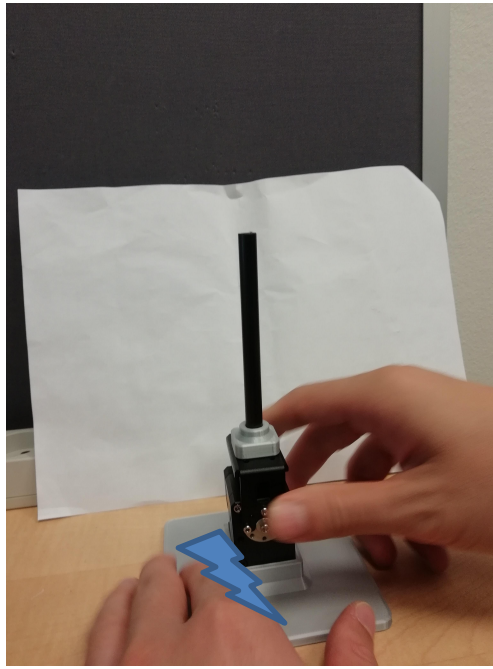




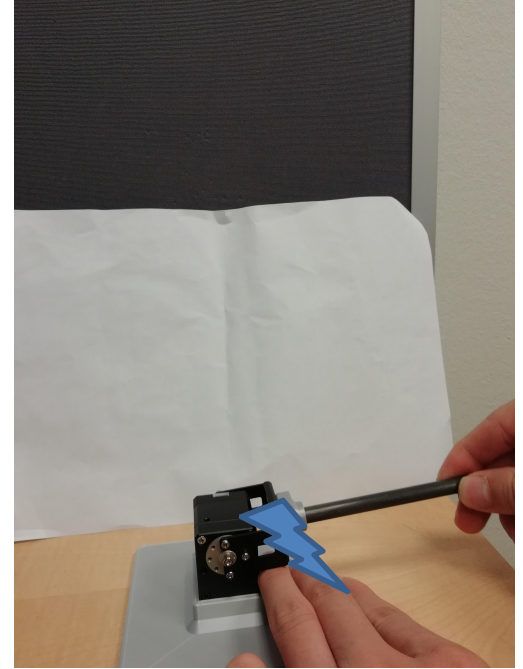
# Exercise safty



**Yes**



**No**



**No**

# Acknowledge and References

Poramate Manoonpong and Cao Danh Do

1. Xiong and Manoonpong, Adaptive Motor Control for Human-like Spatial-temporal Adaptation, IEEE ROBIO, 2018.
2. Fundamentals of Neuromechanics, Valero-Cuevas, Francisco J, 2016
3. DeWolf et al., A spiking neural model of adaptive arm control, the Royal Society B: Biological Sciences, 2016.
4. Spence, Scaling in biology, Current Biology, 2009.
5. Braun et al., Optimal variable stiffness control: formulation and application to explosive movement tasks. Autonomous Robots, 2012.
6. Tassa et al., Control-limited differential dynamic programming, ICRA, 2014.
7. Kronander and Billard, Stability Considerations for Variable Impedance Control. IEEE Trans. on Robotics, 2016.
8. Kadiyallah et al., Generalization in Adaptation to Stable and Unstable Dynamics. PLOS ONE, 2012.
9. Yang et al., Human-Like Adaptation of Force and Impedance in Stable and Unstable Interactions. IEEE Trans. on Robotics, 2011.
10. McIntyre et al., The control of stable postures in the multijoint arm, Experimental Brain Research, 1996.
11. Dorn et al., Predictive simulation generates human adaptations during loaded and inclined walking. PLoS ONE, 2015.
12. D'Avella and Lacquaniti, Control of reaching movements by muscle synergy combinations. Frontiers in Computational Neuroscience, 2013.



# Fill in questionnaire

## Student Feedback Questionnaire

**1. On a scale of 1 to 10, 10 being the highest (i.e., best), how would you rate the teaching method of the teacher?**

(     ) [0 - 10]

**2. On a scale of 1 to 10, 10 being the highest (i.e., best), how would you rate the teacher for the knowledge he possesses about the topic being taught?**

(     ) [0 - 10]

**3. On a scale of 1 to 10, 10 being the highest (i.e., best), how would you rate the interaction skills of the teacher?**

(     ) [0 - 10]

**4. How helpful was the class material (e.g., programming codes) provided to you? (Use 'x' as the tick)**

Email it to **xizi@mmmi.sdu.dk**.

# Thanks for your attention

## Questions

