TU Berlin Robotics WiSe 18/19
Lab Assignment #3

Student Name	**(A)**	**(B1)**	**B2.a**	**B2.b**	**B3.a**	**B3.b**	**B3.c**	**B4**	**B5**
Jiaqiao Peng	x				x	x	x		
Dayyan Smith	x		x	x	x				
Benjamin Oesterle	x		x	x	x	x	x	x	x
Botond Péter Sléber		x	x	x	x	x	x		
Isma-Ilou Sadou	x		x	x	x	x			

Table 1: implementation table

A. Control Theory

1(a) Find the natural frequency ω_n and the natural damping ratio ζ_n of the natural system, i.e., $f = 0$. What kind of system is this?

given: $m = 2,\ b = 12,\ k = 18$

$\omega_n = \sqrt{\frac{k}{m}} = \sqrt{\frac{18}{2}} = \sqrt{9} = 3$

$\zeta_n = \frac{b}{2\sqrt{k \cdot m}} = \frac{12}{2\sqrt{18 \cdot 2}} = \frac{12}{2\sqrt{36}} = \frac{12}{2 \cdot 6} = \frac{12}{12} = 1$

This system is *critically damped* because the damping ratio ζ_n is equal to 1.

1(b) Design a PD controller f that realizes a critically damped system with a closed-loop stiffness of $k_{CLS} = 32$. Assume that the desired position is $x_d = 0$. (Hint: k_{CLS} refers to the stiffness of the closed-loop controller composed of the physical system and the controller f)

The dynamics of the system are described by:

$$m\ddot{x} + b\dot{x} + kx = f \tag{1}$$

where f is the PD-controller we are designing and the right-hand side decribes the physical system (of which the parameters $m = 2, b = 12$ and $k = 18$ are assumed to be known here).

A PD-controller is described by the following control law:

$$f = -k_v\dot{x} - k_p x \tag{2}$$

By inserting (2) into (1) we can derive the closed-loop dynamics:

$$m\ddot{x} + b\dot{x} + kx = -k_v\dot{x} - k_p x \tag{3}$$

which (by simple rearrangement) can also be formulated as:

$$m\ddot{x} + (b + k_v)\dot{x} + (k + k_p)x = 0 \tag{4}$$

Here $k + k_p$ has to be equal to the desired closed-loop stiffness $k_{CLS} = 32$:

$$k + k_p = 32 \iff k_p = 32 - k = 32 - 18 = 14 \tag{5}$$

As we want to achieve critical damping we also know how to choose $b + k_v$:

$$b + k_v = 2\sqrt{m \cdot k_{CLS}} = 2\sqrt{2 \cdot 32} = 2\sqrt{64} = 2 \cdot 8 = 16 \iff k_v = 16 - b = 16 - 12 = 4 \tag{6}$$

Once we have chosen k_p and k_v we can insert their values in our control law (2):

$$f = -4\dot{x} - 14x \tag{7}$$

If all physical parameters are correct, this PD-controller f will realize the desired behavior.

1(c) Assume that the friction changes from linear friction ($b = 12\dot{x}$) to Coulomb friction: $b = 30 \cdot sign(\dot{x})$. Design a controller f that uses a non-linear model-based portion with trajectory following to critically damp the system at all times and maintain a closed-loop stiffness of $k_{CLS} = 32$. Note that f is an m-mass control, while f' is a unit-mass control. What are α, β, k_v and k_p? Write down f' while using the definition of error $e = x - x_d$. (Again: k_{CLS} refers to the stiffness of the closed-loop system composed of the model-based portion and the control-law portion).

The dynamics of the system are still described by:

$$m\ddot{x} + b\dot{x} + kx = f \tag{8}$$

Though our control law this time is defined using control law partioning:

$$f = \alpha f' + \beta \tag{9}$$

where f' (for trajectory following of a unit-mass system) is defined as:

$$f' = \ddot{x}_d - k'_v(\dot{x} - \dot{x}_d) - k'_p(x - x_d) \tag{10}$$

In order for this to work we have to choose α and β in a way that it reduces the system in a way, that it will appear as a unit mass system from the f' input. Given (8) this results in:

$$\alpha = m \qquad\qquad \beta = b\dot{x} + kx \tag{11}$$

Inserting our knowledge from (10) and (11) into our control law partioning (9) and then using this in (8) gives us again the formula describing the closed-loop dynamics of our system:

$$m\ddot{x} + b\dot{x} + kx = m \cdot (\ddot{x}_d - k'_v(\dot{x} - \dot{x}_d) - k'_p(x - x_d)) + b\dot{x} + kx \tag{12}$$

which (by simple rearrangement, using the definition of the error e) can also be formulated as:

$$m\ddot{e} + mk'_v\dot{e} + mk'_p e = 0 \tag{13}$$

Here mk'_p has to be equal to the desired closed-loop stiffness $k_{CLS} = 32$:

$$mk'_p = 32 \iff k_p = \frac{32}{m} = \frac{32}{2} = 16 \tag{14}$$

As we want to achieve critical damping we also know how to choose mk'_v:

$$mk'_v = 2\sqrt{m \cdot k_{CLS}} = 2\sqrt{2 \cdot 32} = 2\sqrt{64} = 2 \cdot 8 = 16 \iff k_v = \frac{16}{m} = \frac{16}{2} = 8 \tag{15}$$

Once we have chosen k_p and k_v we can insert their values in our trajectory control law (10):

$$f' = \ddot{x}_d - 8(\dot{x} - \dot{x}_d) - 16(x - x_d) \tag{16}$$

If all physical parameters are correct, our control law (2) will realize the desired behavior.

1(d) What is the steady-state error $e = x - x_d$ of the system in part (c) (i.e. when $\ddot{e} = \dot{e} = 0$) if it is disturbed by a constant force $f_{dist} = 4$?

Setting all derivatives in the closed-loop dynamics (13) to zero yields:

$$k_{CLS} \cdot e = f_{dist} \tag{17}$$

By rearrangement of this formula we arrive at the steady-state error:

$$e = \frac{f_{dist}}{k_{CLS}} = \frac{4}{32} = \frac{1}{8} \tag{18}$$

B. Visual Servoing

1. Image Features

Feature 1: Isoceles Triangle

Extraction: Only the three edges have to be detected, relatively easy.

Feature Parameters:

- a, b: length of the sides
- u, v: the bounding rectangle sides
- α: rotation of the rectangle ($\alpha \in [0°, 360°]$)

Controllable DoFs: 4 (x, y, z and z-rotation)

Dimensionality of the Image Jacobian: 5×4
($k \times m$ with k = dimension of feature space, m = dimension of task space)

The resulting system of equations is <u>overconstrained</u> (because $k > m$).

Feature 2: Line

Extraction: Only one edge has to be detected, easiest.

Feature Parameters:

- a: length of the sides
- u, v: the bounding rectangle sides
- α: rotation of the rectangle ($\alpha \in [0°, 180°]$)

Controllable DoFs: 4 (x, y, z and z-rotation)
There will be ambiguities regarding z-rotation though!

Dimensionality of the Image Jacobian: 4×4
($k \times m$ with k = dimension of feature space, m = dimension of task space)

The resulting system of equations is <u>well-formed</u> (because $k = m$).

Feature 3: Rectangle

Extraction: Only the four edges have to be detected, relatively easy.

Feature Parameters:

- a, b: length of the rectangle sides
- u, v: the bounding rectangle sides

- α: rotation of the rectangle ($\alpha \in [0°, 180°]$)

Controllable DoFs: 4 (x, y, z and z-rotation)
There will be ambiguities regarding z-rotation though!

Dimensionality of the Image Jacobian: 5×4
($k \times m$ with k = dimension of feature space, m = dimension of task space)

The resulting system of equations is overconstrained (because $k > m$).

2(a) Make sure your implementation is tolerant to adverse camera images and document the steps you did to ensure this.

In real camera images there will be noise. In order to still find a circle, the camera image can be blurred a little. This is already done automatically in the `cv_main.cpp` though.
The last four parameters of the opencv-function `HoughCircles()` are:

1. the upper threshold for the internal Canny edge detector
2. the threshold for center detection
3. minimum circle radius to be detected (0 if unknown)
4. maximum circle radius to be detected (0 if unknown)

Choosing these appropriately might yield better results depending on the exact scenario.

3(a) We need to know the depth distance z of the circle to compute our Image Jacobian. Derive the formula for calculating z and include a short explanation of the derivation in your report.

Based on the exercise slides on perspective projection we know that the ratio of the circle-radius in the image-plane r and the focal length f is the same as the ratio of the circle-radius in the cartesian space $\frac{d}{2}$ and the depth distance z. Given we know f, r and d we can compute z:

$$\frac{r}{f} = \frac{d}{2z} \qquad \Longleftrightarrow \qquad z = \frac{f}{2 \cdot r} \cdot d$$

Parameters:

- r: the radius of the circle in the image plane [in pixels]
- f: the focal length of the camera [in pixels]
- d: the diameter of the real circle [in meters]
- z: the depth distance [in meters]

In reality though this will only serve as an estimate as the detected radius of the circle in the image plane can vary e.g. because of noise or preprocessing steps like gaussian blur).

3(b) In addition to the circle centroid's position, the Image Jacobian also depends on the focal length of the camera. Describe a way to determine this parameter experimentally.

This can be done using the same relationship as in 3(a). We simply place a circle of known size in a fixed distance to the camera and measure the size of its projection in the image plane:

$$\frac{r}{f} = \frac{d}{2z} \qquad \Longleftrightarrow \qquad f = \frac{2z}{d} \cdot r$$

Parameters:

- r: the radius of the circle in the image plane [in pixels]
- f: the focal length of the camera [in pixels]
- d: the diameter of the real circle [in meters]
- z: the depth distance [in meters]

3(c) Specify the image Jacobian $J_I \in \mathbb{R}_{3\times3}$ and describe the equation for each element of the Jacobian in your written report.

$$J_I = \begin{bmatrix} -\frac{f}{z} & 0 & \frac{u}{z} \\ 0 & -\frac{f}{z} & \frac{v}{z} \\ 0 & 0 & \frac{f \cdot d}{z^2} \end{bmatrix}$$

Parameters:

- f: the focal length of the camera [in pixels]
- z: the estimated depth (cf. 3(a)) of the circle [in meters]
- (u, v): center coordinates of the circle in the image plane [in pixels]
- d: the diameter of the real circle [in meters]