Virtual Reality
Winter Term 2018/2019

Assignment Sheet 4

A sophisticated interaction technique is strongly dependent on the properties of the input device and the applied transfer function (see Figure 1). Input devices can provide isotonic, elastic or isometric input. Examples of transfer functions, on the other side, include position, rate and acceleration control. In this assignment, you will work with a mouse as a representative of isotonic input, which measures relative movements in two directions. The spacemouse device provides elastic input for six degrees of freedom.

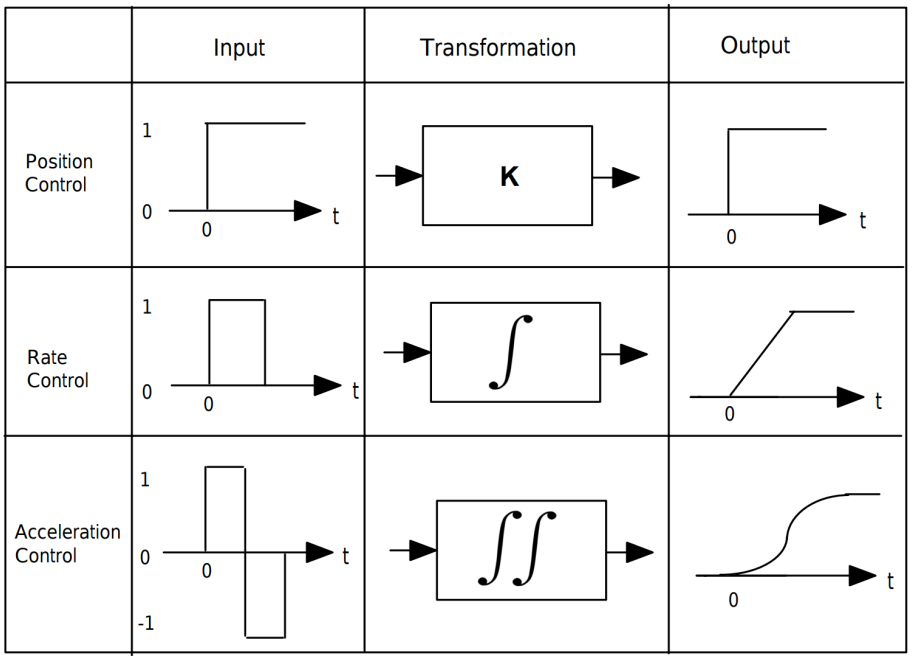

Figure 1: Idealized control inputs (left column) for obtaining step changes in output level (right column) for position, rate and acceleration control[1]

[1] Zhai, Shumin. Human performance in six degree of freedom input control. University of Toronto, 1996.

The goal of this assignment sheet is to implement a spatial manipulation task in which multiple target objects have to be selected and moved with a virtual hand using *isotonic* and *elastic* input devices in combination with the three mentioned transfer functions (see Figure 2). The resulting six combinations are assigned to the number keys 1 to 6 in the provided framework.

Figure 2: Spatial manipulation task to be implemented in this assignment sheet. Objects in close proximity of the virtual hand are highlighted and can be dragged to another location.

The class `ManipulationManager` in `Manipulation.py` initializes and administrates the six manipulation techniques. Each individual technique is derived from the base class `Manipulation`. The respective behavior of each technique is implemented in its `manipulate()` function, which is executed every frame. As a starting point, the class `IsotonicPositionControlManipulation` is already given and serves as a reference implementation for the remaining techniques.

Important Note: Before you start working at a computer of our lab, please investigate if your spacemouse model shines blue when connected to your computer. If yes, set the variable `SPACEMOUSE_TYPE` in `SpacemouseType.py` to `'blue'`. If no blue light is present, set the variable to `'old'`. If you are not sure whether your device fires correct input, add `print(_x)` to line 90 of `lib/Device.py` and move the device during runtime.

For your submission, please pack your exercise folder into a `*.zip` file and upload it to Moodle. Your are required to do so by **06 December 2018, 11:55 PM**. Group work in pairs of two is permitted. You will present and explain the submitted code in the lab class on **07 December 2018**. This assignment sheet contains tasks worth **15 points** and will be weighted by **20%** for your total lab class grade.

Exercise 4.1 (2 points)

The class `IsotonicPositionControlManipulation` is already implemented such that the virtual hand can be moved using isotonic position control. Objects in close proximity of the hand are highlighted. Implement a dragging functionality such that the highlighted objects can be moved when the mouse button is pressed. For this purpose, implement the three functions `start_dragging()`, `object_dragging()` and `stop_dragging()` in the class `ManipulationManager`. We have added an additional helper field `DraggingOffsetMatrix` to each `TriMeshNode` that can be used to temporarily store a transformation matrix, e.g. `node.DraggingOffsetMatrix.value = ...`.

Exercise 4.2 (4 points)

Implement the classes `IsotonicRateControlManipulation` (key 3) and `IsotonicAccelerationControlManipulation` (key 5) such that the virtual hand can be moved using isotonic rate control and isotonic acceleration control. Adjust the input scaling factors or insert additional helper variables if required.

Exercise 4.3 (6 points)

Implement the classes `ElasticPositionControlManipulation` (key 2), `ElasticRateControlManipulation` (key 4) and `ElasticAccelerationControlManipulation` (key 6) such that the virtual hand can be moved using elastic position control, elastic rate control and elastic acceleration control. Adjust the input scaling factors or insert additional helper variables if required.

Exercise 4.4 (3 points)

Find a suitable use case for each of the six combinations and write them in a new text file within the exercise folder. You can think of examples in the real world or within the scopes of object manipulation and viewpoint navigation in virtual environments.