Software and Tools for Computational Engineering

RWTH AACHEN UNIVERSITY

Software Lab Computational Engineering Science

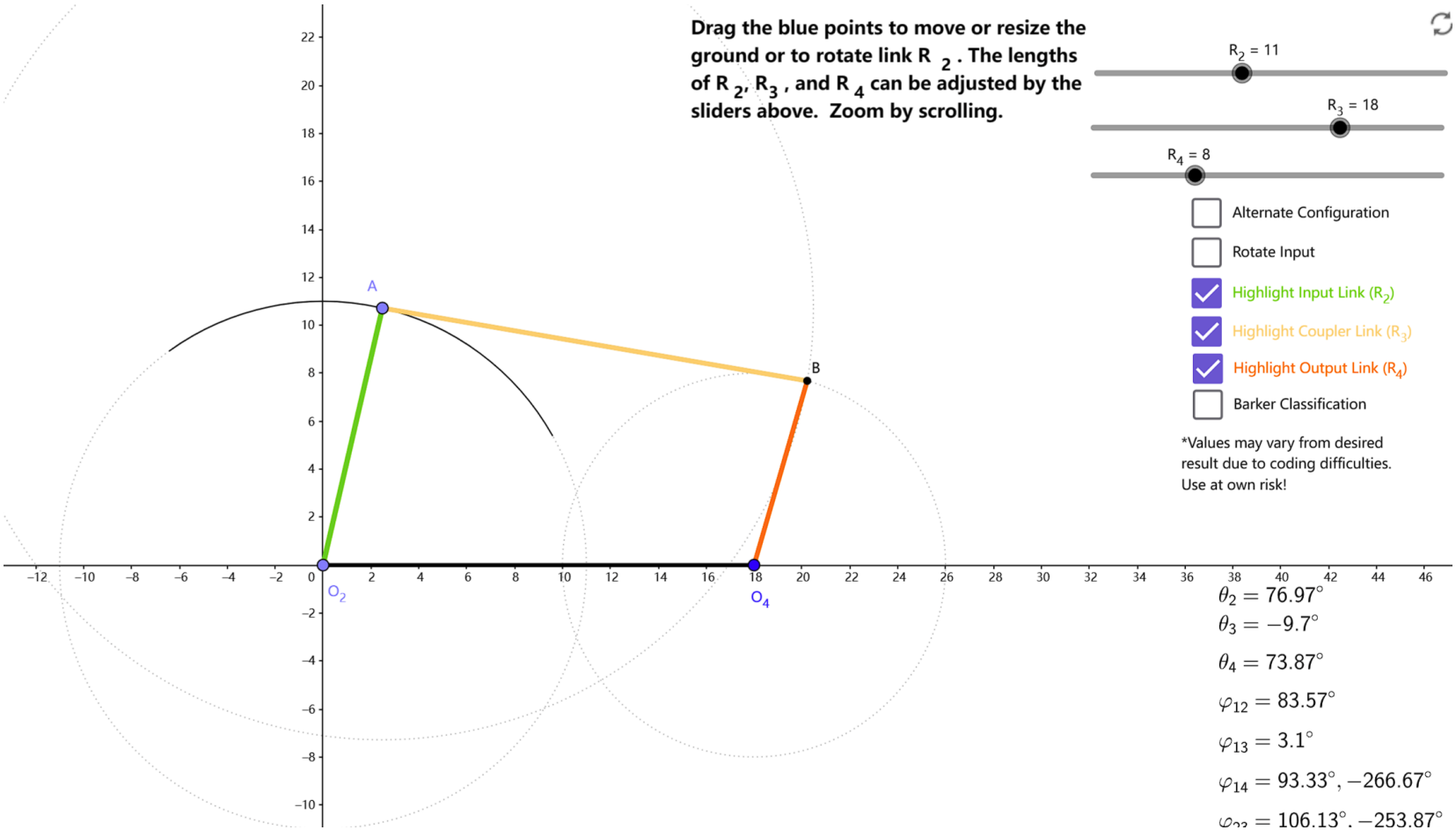
Group 12, Pusher Mechanism

Aaron Floerke, Arseniy Kholod, Xinyang Song and Yanliang Zhu

Informatik 12: Software and Tools for Computational Engineering (STCE)
RWTH Aachen University

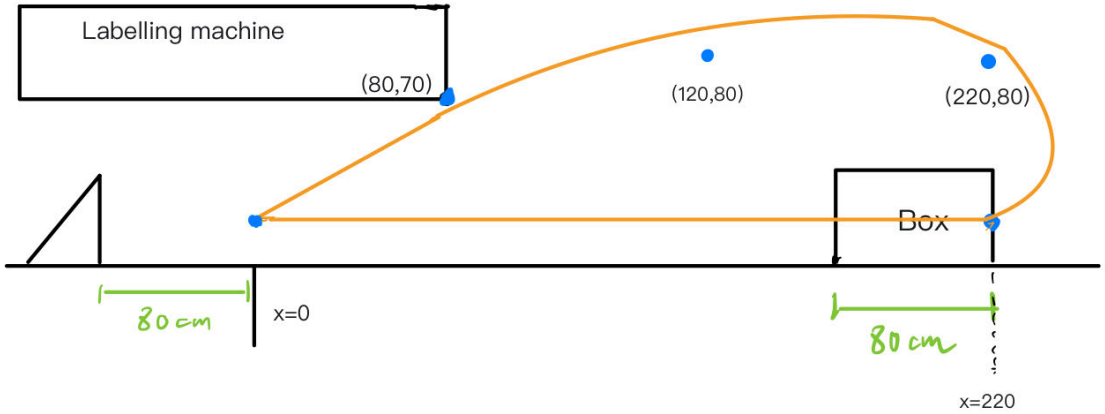
Contents

- Online Demo: `https://www.geogebra.org/m/BueCG9ch`



User Requirements

- Simulate the motion of the pusher mechanism.
- Implement all motion types for the four-bar linkage with one bar fixed (8 cases according to `https://en.wikipedia.org/wiki/Four-bar_linkage`).
- Implement GUI with motion animation and the ability to choose geometrical parameters.
- Solve an optimization problem.



- Find suitable parameters for the pusher mechanism to meet requirements:
  - Push box with size 80×60 from $x = 220$ to $x = 0$
  - Do not cross the area of the labelling machine (Area with $x < 80$ and $y > 70$).
  - Pass above points $(120, 80)$ and $(220, 80)$
- Provide the following data for the designed four-bar mechanism:
  - Position of the two fixed pivot positions
  - Lengths of the three moving links and of the fourth base link
  - The position of the couple offset point relative to the coupler

*all coordinates and length are in cm

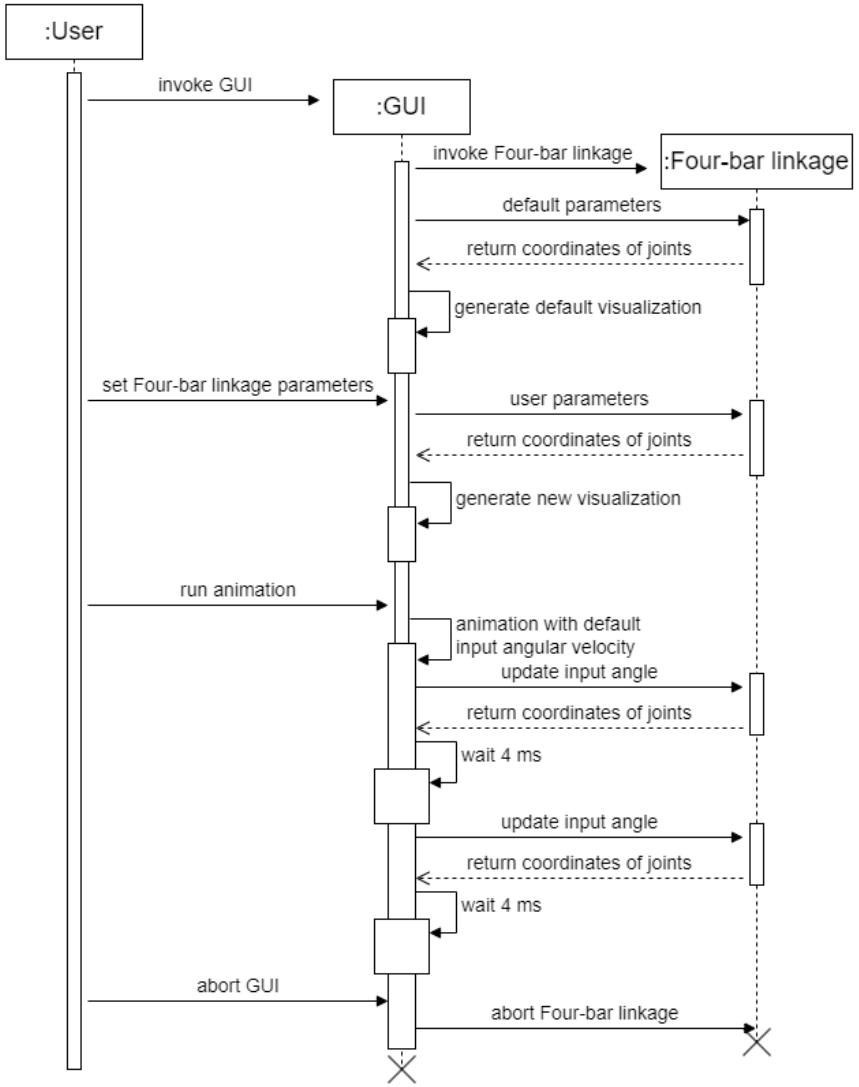
- **Four-bar linkage model**:
  - System simulates the motion of the four-bar linkage.
  - System distinguishes between all different motion cases.
  - System does not crash with any input of geometrical configuration.
  - System provides asserts or/and exception mechanisms to validate input data.
- **Tests**:
  - Implement test cases to cover all motion cases.
  - Provide reference data to compare results with.
  - Implement test cases with bad input to test system stability.

- **Graphical User Interface**:
  - GUI provides the four-bar linkage visualization.
  - User can input geometrical data by moving a point on a slide bar or/and using a keyboard.
  - GUI reacts to new input data by changing the four-bar linkage visualization accordingly.
  - GUI provides animation of the four-bar linkage motion.
  - User can choose parameters of this motion, e.g. restrict rotation angles.
  - User can move linkage by pulling an input bar using a computer mouse.
  - GUI is coupled with the four-bar linkage model to use implemented motion cases for animation.
- **Optimization problem**:
  - It should be possible to find a solution (manually) for the optimization problem using the four-bar linkage model.
  - GUI visualizes the solution.

- **Performance**:
  - The four-bar linkage model is fast enough to provide smooth GUI animations.
  - GUI animations are not slower than 30 frames per second.
- **Usability**:
  - Every essential part of the four-bar linkage model is well documented.
  - GUI is easy to operate.
  - GUI functionalities are well documented.
  - All changeable parameters are explained.
  - GUI animation displays trajectories of bar connection points.
- **Locality**:
  - GUI and the four-bar linkage model both can run on a local machine.
  - Additionally: Local GUI should be able to connect to a remote four-bar linkage model.
  - Additionally: It should be possible to provide a web GUI.

Prototype

Sequence diagram for animation



Gantt Chart

Summary and Conclusion

- **User requirements**
- **Optimization problem**
- **System requirements**
- **Prototype**:
  - Sequence diagram for communication between user, GUI, and Four-bar linkage model.
  - Graphical user interface visualization.
- **Project with Gantt Chart**
- **Next tasks**:
  - Design classes, interfaces, functions.
  - Design detailed use-cases.
  - Choose languages and environment to implement Four-bar linkage model and GUI.
  - Implementation and documentation.
  - Implement tests and use-cases.
  - Solve the optimization problem.