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

Preface

Four-bar linkage model
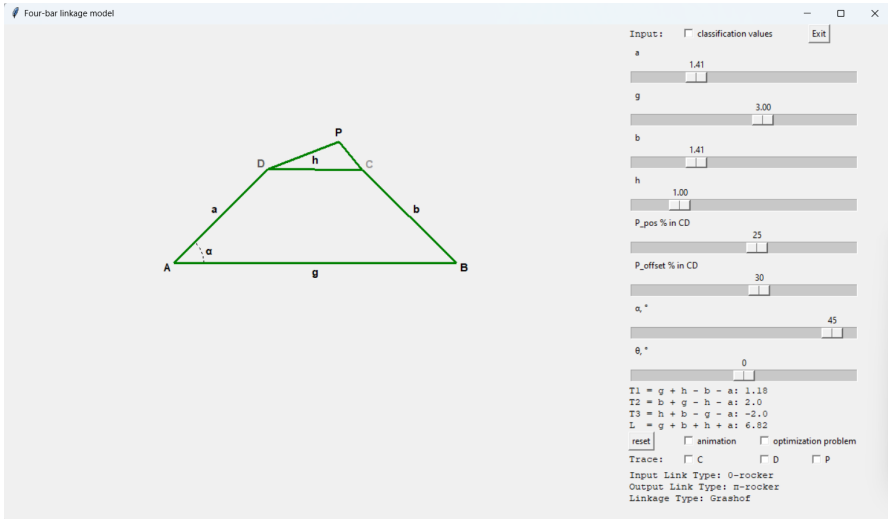
- Implement 27 motion types of the four-bar linkage with one bar fixed:
  - Classification values:
    - $T_1 = g + h - b - a$
    - $T_2 = b + g - h - a$
    - $T_3 = h + b - g - a$
- Implement GUI with motion animation and the ability to choose geometrical parameters:
  - Length of the bars
  - Position of the coupler
  - Input angle
  - Angle relative to the horizon
  - Classification values as alternative input

No.	T_1	T_2	T_3	$T_1 T_2$	$T_1 T_3$	a	b
1	+	+	+	+	+	crank	rocker
2	0	+	+	0	0	crank	π–rocker
3	-	+	+	-	-	π–rocker	π–rocker
4	+	0	+	0	+	crank	0–rocker
5	0	0	+	0	0	crank	crank
6	-	0	+	0	-	crank	crank
7	+	-	+	-	+	π–rocker	0–rocker
8	0	-	+	0	0	crank	crank
9	-	-	+	+	-	crank	crank
10	+	+	0	+	0	crank	π–rocker
11	0	+	0	0	0	crank	π–rocker
12	-	+	0	-	0	π–rocker	π–rocker
13	+	0	0	0	0	crank	crank
14	0	0	0	0	0	crank	crank
15	-	0	0	0	0	crank	crank
16	+	-	0	-	0	π–rocker	crank
17	0	-	0	0	0	crank	crank
18	-	-	0	+	0	crank	crank
19	+	+	-	+	-	0–rocker	π–rocker
20	0	+	-	0	0	0–rocker	π–rocker
21	-	+	-	-	+	rocker	rocker
22	+	0	-	0	0	0–rocker	crank
23	0	0	-	0	0	0–rocker	crank
24	-	0	-	0	+	0–rocker	0–rocker
25	-	-	-	-	-	rocker	crank
26	0	-	-	0	0	0–rocker	crank
27	-	-	-	+	+	0–rocker	0–rocker

Figure from "Classification, geometrical and kinematic analysis of four-bar linkages" 10.15308/Sinteza-2018-261-266 by Ivana Cvetkovic et al.
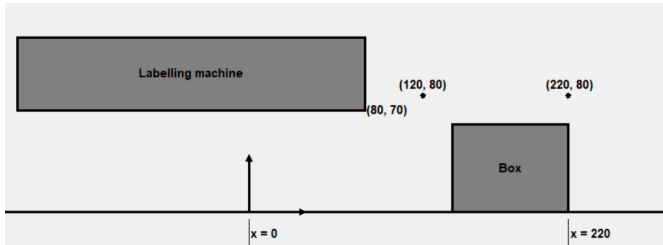
▶ Solve an optimization problem:
  ▶ Push box with size 80×60 from $x = 220$ to $x = 0$
  ▶ Do not cross the area of the labelling machine (Area with $x < 80$ and $y > 70$).
  ▶ Pass above points $(120, 80)$ and $(220, 80)$

- **Four-bar linkage model**:
  - System simulates all the motion types of the four-bar linkage.
  - System does not crash with any input of geometrical configuration.
- **Tests**:
  - Implement test cases for geometry.
  - Implement test cases with bad input to test system stability.
- **Graphical User Interface**:
  - GUI provides the four-bar linkage visualization and motion animation.
  - User can input geometrical data by moving a point on a slide bar.
  - GUI is coupled with the four-bar linkage model to use implemented motion cases for animation.
  - GUI provides tracing for trajectories of the points.
  - GUI classifies of the linkage.
- **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 and all functionalities are self-explanatory.

    ▶ GUI source code is well documented.

- ▶ **1. Operating System:**
  - ▶ Xubuntu/Windows
- ▶ **2. Developing Environment:**
  - ▶ Programming Language: Python.
  - ▶ IDE: Spyder/Pycharm.
  - ▶ Package Manager: Anaconda.
- ▶ **3. Libraries:**
  - ▶ Frontend: tkinter, math, numpy
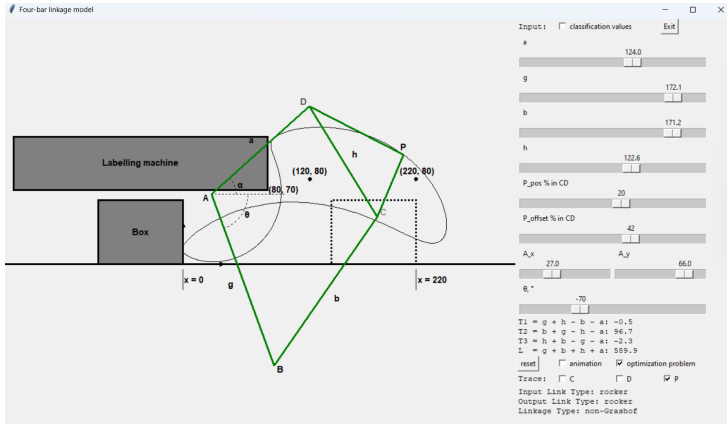 - ▶ Backend: math, numpy
- ▶ **4. Version Control System:**
  - ▶ GitHub: Remote code repositories for team collaboration, code reviews, and version control.
    `https://github.com/einsflash/Project_Pusher_Mechanism`
- ▶ **5. Frameworks:**
  - ▶ Pdoc: Used for generating project documentation, helping the team understand and maintain the code better.
  - ▶ Makefile: For build management.

- ▶ 9 degrees of freedom (all lengths in cm):
    - ▶ Length of four bars: $a = 124.0$, $b = 171.2$, $g = 172.1$, $h = 122.6$.
    - ▶ Coupler position: $P_{pos} = 20.0\%$, $P_{offset} = 42.0\%$ of h.
    - ▶ Position of point A: $A_x = 27.0$, $A_y = 66.0$.
    - ▶ Angle of ground bar relative to horizon: $\theta = -70.0°$

Documentation for Frontend(GUI)

API Documentation

gui

• View Source

class GUI: • View Source

linkage

tk

width

height

model_frame

model_animation

toolbar_frame

def trace_C(self):   • View Source

def trace_D(self):   • View Source

def trace_P(self):   • View Source

positions_C

positions_D

positions_P

x_axis

y_axis

A_x

A_y

pin_box_to_coupler

prev_coupler_position

prev_box_position

def init_toolbar(self):   • View Source

def init_linkage_display(self):   • View Source

Documentation for Backend

API Documentation

class FourBarLinkage
- FourBarLinkage()
- AB
- BC
- CD
- DA
- alpha
- theta
- alpha_rad
- theta_rad
- coupler_position
- coupler_offset
- t
- alpha_velocity
- C_mode
- init_default_values()
- run()
- check_Parameter()
- find_Linkage_Type()
- calculate_Classification_Value()
- calculate_Edge_Value()
- calculate_alpha_lims()
- calculate_Point_Position()
- calculate_C_Position()
- calculate_P_Position()
- animation_alpha()
- switch_C2_C1()

built with pdoc

four_bar_linkage

» View Source

class FourBarLinkage: » View Source

```
FourBarLinkage(
    AB,
    BC,
    CD,
    DA,
    alpha,
    theta,
    coupler_position,
    coupler_offset,
    timeinterval,
    alpha_velocity
)                                                       » View Source
```

AB

BC

CD

DA

alpha

theta

alpha_rad

theta_rad

coupler_position

coupler_offset

t

alpha_velocity

C_mode

def init_default_values(self):   » View Source

def run(self):   » View Source

► Cvetkovic, Ivana and Stojicevic, Misa and Popkonstantinović, Branislav and Cvetković, Dragan. (2018). Classification, geometrical and kinematic analysis of four-bar linkages. 261-266. 10.15308/Sinteza-2018-261-266.