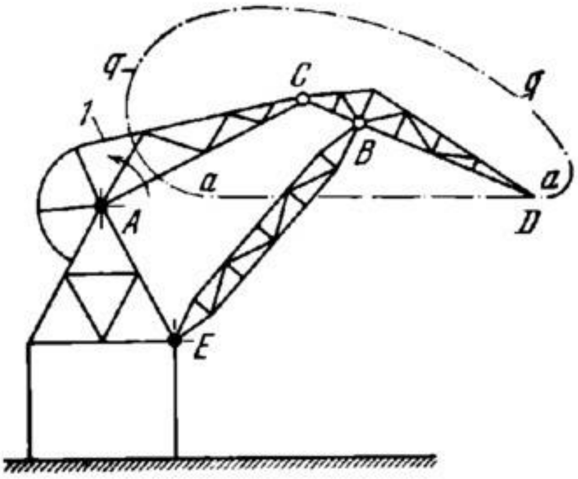
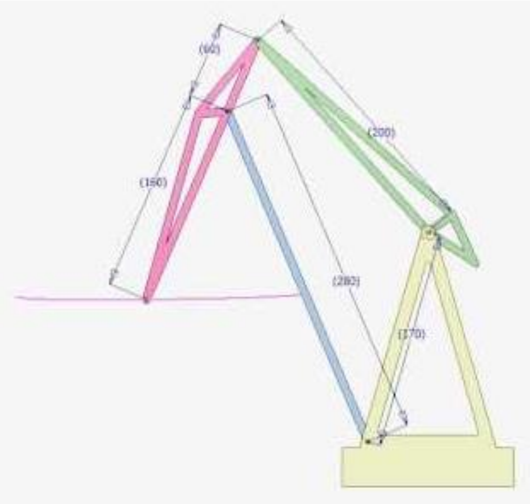
Project 1: Harbor crane

Some harbor cranes utilize the kinematic structure of an approximated straight-line mechanism (usually, a four-bar mechanism with appropriately chosen lengths of the links) to transport the payload horizontally. The project is focused on using multibody simulations to establish some critical design parameters of a crane.





Tasks:

1. Browse the internet or library and choose your favorite crane. Estimate its dimensions and possibly masses of the links and the lifting capacity.
2. Create a multibody model of the crane, check the straight-line motion range, and verify how much the straight-line is approximated.
3. Parameterize your model and optimize it (by altering dimensions of its links) to obtain a better approximation of the straight line, a larger motion range, or both. Keep dimensions of the mechanism reasonable, i.e., introduce some design constraints in your optimization.
4. Ideally, to make the crane operation more intuitive, the constant velocity of the motor should result in the constant horizontal velocity of the payload. A constant "gearing ratio" is unattainable for the crane kinematic structure; however, we can be closer or farther away from the constant velocity requirement.

   The straight-line requirement and the constant velocity requirement are quite contradictory goals. Nevertheless, we can try to find a reasonable compromise between them. Your task is to run another optimization and find this compromise.
5. Figure out what is the balancing mass for. Add a balancing mass to your crane; make sure it fulfills its duty while not being too large.
6. Have fun.

Project 2: Parallel manipulator

Designers, to reach good decisions, must somehow estimate the expected performance of the mechanism under construction. When the mechanism is complicated, which is the case when we deal with parallel manipulators, multibody methods prove to be very useful to investigate the design, especially at its preliminary stage. This project is focused on creating and investigating a multibody model of an industrial robot.

Tasks:

1. Find a parallel robot that you want to model and simulate. Learn about its kinematic structure, crucial dimensions, and masses. Other information, e.g., maximum velocities and accelerations, joint limits or workspace shape, parameters of motors, etc., would also be welcome.
2. It is advised—however, not strictly required—that the robot's kinematics allows for separate analysis of translational and angular motion – the goal is to start modeling with three translational DOFs and then supplement the model with the other three DOFs. For some structures, this cannot be done, and all six DOFs must be taken into account simultaneously.
3. Create a multibody model of the robot (or its translational DOFs only). Focus on kinematics and make it possible to impose driving constraints alternatively on the motors or on the end-effector.
4. Propose an end-effector test trajectory (or a set of trajectories) typical for the robot duties. You can start from a straight line motion with an appropriately chosen velocity profile and then think about more complicated paths. Try to make the trajectory demanding so that the robot must engage its resources to the full extent (i.e., the velocities and accelerations should be close to the maximum ones, the path should go through a large part of the workspace, etc.)

5. Utilizing the test trajectories, perform necessary simulations. Your goal is to determine the required motor torques as well as velocities and accelerations.
6. When you succeed with translational motion, repeat the procedure for all DOFs of the robot.
7. Assume/estimate the stiffness of the motors. Check how it is reflected by the end-effector stiffness at various points of the workspace. Find natural frequencies of the mechanical system.
8. Account for the links' flexibility. Check how it affects the end-effector stiffness and natural frequencies.
9. Try to enjoy the project.

Project 3: Counterweight trebuchet

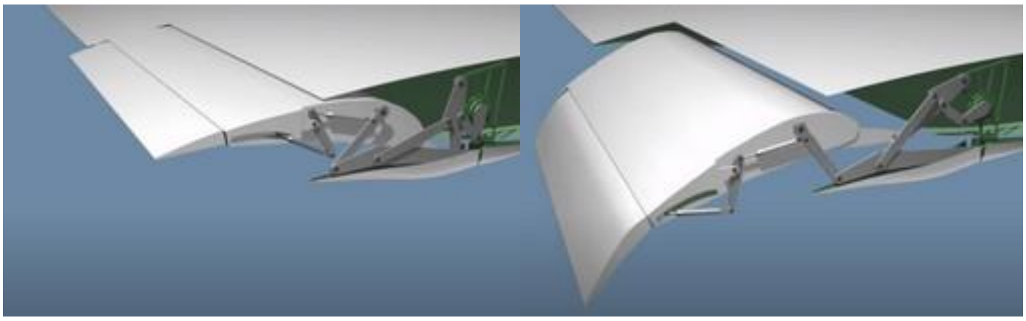
Multibody dynamics simulations make it possible to analyze and optimize a mechanical design. Your task will consist in modeling and optimizing a trebuchet – medieval siege engine capable of throwing projectiles at significant distances.

Tasks:

1. Go to the library or browse the internet in order to select the trebuchet variant that you want to investigate. Make reasonable assumptions regarding the counterweight and projectile masses; estimate the machine dimensions.
2. Create a multibody model of the trebuchet and parameterize its geometry. For the initial tests, the model can be vastly simplified. Perform some simulations to check whether the model works properly.
3. Select design parameters, and perform a design study to learn how individual design parameters affect the range of projectile throw.
4. Decide on design constraints and formulate the goal function (most probably, you will optimize the weapon range). Then, optimize the machine.
5. Make the model more realistic, e.g., introduce friction (where applicable), aerodynamic drag force (if not modeled previously), try to reproduce the sling-release mechanism in greater detail, account for flexibility of the arm, etc.
6. Optimize the machine again.
7. Propose improvements to the traditional trebuchet design and validate them via simulation.
8. Try to prove that modern science can successfully compete with medieval craftsmen's skills.

Flap extraction/retraction

see for example: https://www.youtube.com/watch?v=kKDMjc3l_gw



Combination of planar 4-bar mechanisms, actuated by one angular eletric motor

- model the mechanical system using rigid bodies and ideal revolute hinges in Matlab (e.g. using dedicated formulas, or a simple multi-body scheme) and study its kinematics
- model the mechanical system in a multibody solver (e.g. ADAMS, MBDyn or other) and verify the kinematics of the simple model
- model some simplified aerodynamic load (either constant, or configuration-dependent, e.g., a pressure acting of the surfaces proportional to their orientation; the corresponding force increases with the amount of exposed surface); evaluate the torque required for extraction/retraction for constant angular velocity of the driving link
- simulate the extraction/retraction using an equivalent model of the compliance and transfer function of the motor
- (extra): size a DC electric motor for the required torque, model its dynamics and simulate flap extraction/retraction by prescribing the desired angular velocity of the driving link and determining the required voltage input.
- (extra): consider friction in joints and slides

Landing gear extraction/retraction

see for example: https://www.youtube.com/watch?v=cm0rvntP2tE

Several geometries and configurations are used. In some cases, the motion is not planar, but can be handled accordingly, at the cost of a slightly greater complexity.

Combination of planar 4-bar & 5-bar mechanisms, actuated by one linear hydraulic actuator

- model the mechanical system using rigid bodies and ideal revolute hinges in Matlab (e.g. using dedicated formulas, or a simple multi-body scheme) and study its kinetostatics
- model the mechanical system in a multibody solver (e.g. ADAMS, MBDyn or other) and verify the kinematics of the simple model
- model some simplified aerodynamic load (constant); evaluate the force required for extraction/retraction for constant linear velocity of the actuator
- simulate the extraction/retraction using an equivalent model of the compliance and transfer function of the motor
- (extra): size the actuator and simulate the retraction/extraction with constant prescribed feed and discharge pressure
- (extra): consider downstops engagement after extraction/disengagement before retraction

Solar array deployment

https://www.esa.int/ESA_Multimedia/Images/2019/06/Solar_Orbiter_array_deployment_test

- model the system as a chain of rigid bodies, connected by revolute hinges, preloaded springs, and dampers (lumped compliance), by writing a simple Matlab model, and analyze the transition from initial to intermediate configuration (transient dynamics)
- model the system in a multibody solver (e.g. ADAMS, MBDyn or other) and verify the results of the simple model
- include in the model an equivalent compliance and transfer function model of the motors that are used for transition from intermediate to final configuration, the downstops, and simulate the completion of the deployment
- (extra): size the DC electric motors and simulate the completion of the deployment by prescribing the desired angular velocity pattern and analyzing the required control voltage
- (extra): make panels flexible (e.g., shells, or beams in 2D); compare with rigid model