The DLR Cables Library

Tobias Bellmann[1] Andreas Seefried[1] Thomas Bernhofer[1]

[1]Institute of System Dynamics and Control, German Aerospace Center, Germany,
{firstname,lastname}@dlr.de

Abstract

The DLR Cables Library can be used to simulate steel cables with nonlinear stiffness such as steel wire cables with and without non-metal core or coatings. The cable itself is simulated as discrete element component, and can easily be connected to Modelica Multibody components.
Additional components like winches and pulleys are included in the library. With those, complex cable systems with multiple interconnected pulleys become possible in Modelica. Two applications of the library are demonstrated, a construction crane and a cable robot used to compensate gravitation in a space robotic application.
Keywords: Modelica, Cables, Pulleys, Winches, Ropes

1 Introduction

In technical applications, cables oftentimes play important roles e.g. in load handling, transportation, robotics, etc. While the Modelica Standard library contains a feature-rich set of multi-body components (Otter, Elmqvist, and Mattsson 2003), the simulation of cable systems with pulleys and winches right now is only possible with additional libraries (e.g. (Wu et al. 2023), (Börner 2016), (Berger and Heinrich 2011)). In this paper, we would like to present a pure Modelica language based implementation of nonlinear cables, pulleys and winches (see example in figure 1). The components can be coupled with the standard Modelica Multibody frame connectors, enabling the integrated simulation or optimization of complex mechanical systems. The implementation of the cables is focused on simulation speed, allowing complex models, e.g. a construction crane, to be simulated in real-time. However, this library is mainly intended to model the (nonlinear) stretching behaviour of steel or composite cables both of free cables under load and cables wound up on a winch. The bending behaviour of the cables is simulated, but due to the used modeling approach and the highly nonlinear nature of the inner cable mechanics, this can result in sub-par accuracy concerning bending stiffness, damping, eigenmodes, etc.

1.1 Features of the library

The DLR Cables library provides models for cables, winches and pulleys, as well as models to connect cables to multi-body frames. The structure of the library can be seen in Figure 2. A cable consists of `n` discrete cable elements with flexible length.

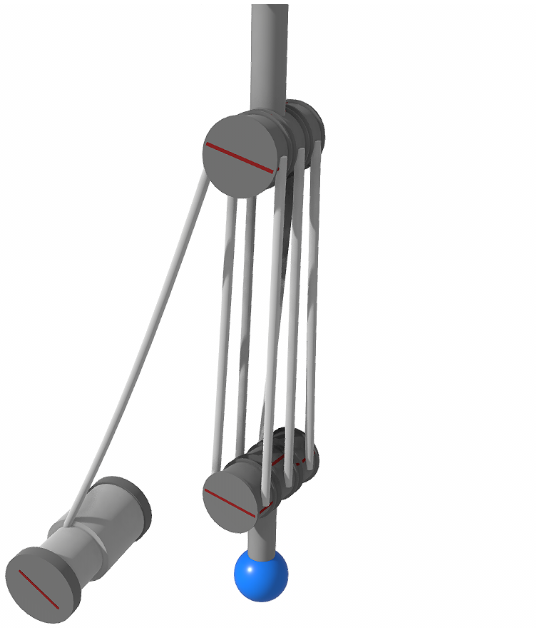

Figure 1. A triple block and tackle system to lift heavy masses, simulated with the DLR Cables Library.

Each cable element has two connectors, a `cableFrame_a` and `cableFrame_b`. In `cableFrame_b`, the normalized direction of the current cable segment is stored and available for the next cable element to calculate the bending of two consecutive cable elements. The cable stiffness is defined by a forces table: by lengthing the cable, a counterforce between `cableFrame_a` and `cableFrame_b` is applied. Thus, a nonlinear stiffness behaviour can be easily implemented. The table has two columns: first the lengthening of a 1m cable, second the force that is acting against that lengthening. In addition to the stiffness force, a damping force acts against the velocity of the cable element along its direction. The damping coefficient is constant. For bending, a constant stiffness and constant damping coefficient can be set. It is possible to tear a cable. The implementation is as follows: the force between `cableFrame_a` and `cableFrame_b` is set to zero. So there is still a complete cable, but one cable element can be streched without any counterforce. For visualization, the cable's transparency element that is 'broken' is set to 100%.

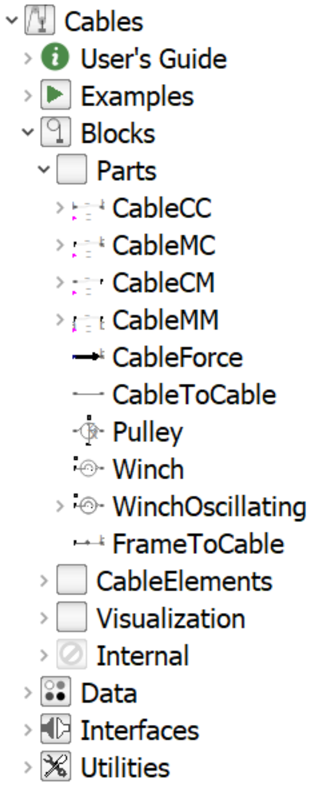

Figure 2. Package overview, showing the main components of the DLR Cables Library.

1.2 Cable models

The Cable models utilises a FEM approach to model a cable with tensile and bending stiffness / damping. Torsion forces are not yet implemented. Three cable models are provided, differing only in their interfaces (Cable-to-Cable, Multi-Body-to-Cable, Cable-to-Multi-Body). The cable stiffness can be defined via a table to reflect non-linear stiffness characteristics of e.g. cables with a non-metal core or coatings. The bending stiffness depends on the modulus of elasticity and the diameter of the cable. A simplified model is used to reduce the computational effort.

1.3 Winch models

The winch model can be used to simulate a winch connected to a multi-body frame. There are two different models available. An ideal winch that provides as much cable as needed and a more realistic oscillating winch where the cable is wound up and unwound in a meandering manner. Both convert the rotation of the winch flange into an elongation or shortening of the attached cable. It is possible to store the cable stresses during the windup of the cable to model the varying elongation of the cable under dynamic loads. When unwinding the winch, this elongation is restored.

1.4 Pulley model

The pulley model allows the connection of two cables to model a pulley system. If one cable is elongated, the other one is shorted by the same length. Effects of asymmetrical stresses and elongations can be considered. For instance, if an elongated cable under load is winded up the cable and the rotational angle of the pulley are large than the transported un-elongated cable. If it is unwinded on the other side of the pulley, a different length of cable is avaliable, because the stress on the cable changes.

2 Modeling approach

In order to achieve a fast simulation speed, several methods are used to simplify the mechanics of the cable system components. The concepts and ideas behind the modeling approach are detailed in this section.

2.1 Basic concepts of the cable model

Every cable consists of n cable segments. Each segment has a variable length and is comprised of two masses, interconnected with a spring damper system to handle tensile strength and lengthening, see Figure 3. The masses are positioned at the ends of the cable segment, avoiding unnecessary transformations from the pose of the connectors to the location of a single central mass. By separating the segments' mass into two separate point masses with half the weight $m/2$, the calculation of rotational inertia of a single body can be omitted, further reducing complexity (but of course by introducing errors such as ignorance of rotations in longitudinal directions of the cable).
In general, the cable segment dynamics is only defined by forces on the two separate masses. Torques (e.g. caused by bending stiffness) are generated by a pair of opposite forces f_{bend} on the two masses of the segment. In order to calculate effects caused by bending of the cable, the angle between the two segments has to be known, therefore the direction of the cable segment is communicated to the adjunct segments. All forces are calculate in the world frame.

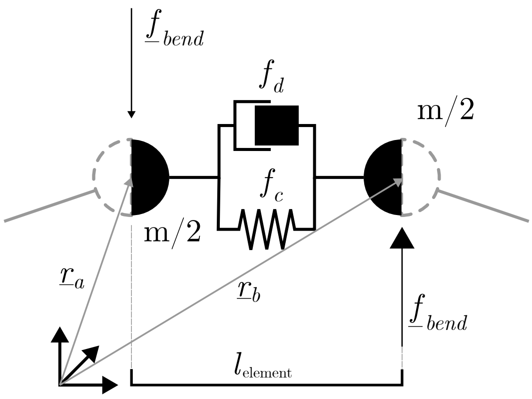

Figure 3. A single cable segment with point masses at the ends and acting forces.

2.2 Basic concept of winches and pulleys

By default, a cable has a fixed length, only subject to change under tensile forces. However, technical systems

DOI
10.3384/ecp21727

oftentimes use winches to unroll or roll up the cable. While it is possible to use forces to bend a FEM based cable around a winch drum, this approach is very costly in terms of calculation time. In order to achieve fast simulation results, suitable for real-time and optimization scenarios, the winch concept in this library differs from this approach: Each cable has a variable length and mass, and it is possible to control length and mass changes from the cable connector. This allows for an external component, in this case winches and pulleys, to change the length of a cable, even if there is no actual unwinding process. Of course, this simplification introduces several modeling errors, such as an unrealistic mass transport behavior and a constant mass and inertia of the winch. The latter is mitigated by also varying the mass of the winch to account for the mass transfer between the winch and the unrolling cable.

2.3 Elongation of a cable on a winch

In many applications where a load is transported with a winch, the load acting on the cable might change during operations. One example for this is an elevator, where the load changes with the exchange of passengers. A noticeable effect is the persistence of the lengthening of a cable under load when rolled up on the winch due to friction forces on the winch. Therefore, when modeling a simplified winch, the elongated state of the cable has to be preserved during the roll up of the cable, and restored during the unroll process. This can be implemented e.g. with a look-up table or other kinds of storage systems.

3 Implementation

The Modelica language allows for a distributed, object-oriented modeling approach, therefore the library is structured in models and sub-models explained in this chapter.

3.1 Interfaces

As explained in the previous section, the concept of the DLR Cables Library reduces the complexity of a full multi-body modeling approach to an point mass based series of cable segments. Therefore, the standard Multibody Frame interfaces cannot be used, and a special `CableFrame` connector without orientations and torques is introduced:

Listing 1. CableFrame connector

```
connector CableFrame
  SI.Position r_0[3];
  Real e_pre[3] "The normalized direction
      of the last cable element in world
      coordinates";
  Real e_next[3] "The normalized direction
      of the next cable element in world
      coordinates";
  flow SI.Force f[3]
    "Cut-force resolved in WORLD frame.";
end CableFrame;
```

It is notable, that the direction of the previous and next cable segment is also communicated with this connector. This enables the calculation of the bending stiffness and damping.

3.2 Cable

The cable model comes in three variations, only differing in the available connectors (see figure 4). The models can either be connected with other cables or Modelica MultiBody frames on either side. Internally, the ca-

Figure 4. The three available cable models (from left to right): CableCC, Cable MC, CableCM.

ble models hold an array of n `CableSegment` models, representing the cable segments. Only the first and last `CableSegment` models are connected to the outside world, the other $n-2$ segments are connected to each other in a for-loop connect statement. Other tasks of the

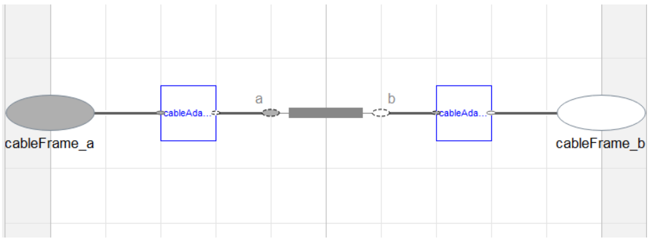

Figure 5. Cable model with two cable connectors (can be connected to other cables, pulleys or winches). In the middle of the diagram, the model array for the n cable segments is visible.

Cable models are the initialization of the cable segment positions, the calculation of the elongation under load and the calculation of the overall cable length. Initially the cable has an overall lenght of l_{cable}. Additionally, the cable has an input Δl, defining the length change of the cable caused by pulleys and winches. The cable is parameterized using the following parameter record:

```
record CableData "base class to hold cable
    information for 1m cable"
  parameter String Name "Name of cable";
  parameter Real forces[:,2]
    "Linear approximation of force due to
        stiffness of the cable of 1m.
        Current force will be calculated
        with [del_l1[m], Force_1 [N];
        del_l2[m], Force_2 [N] ].";
  parameter Real d_axial(unit="N.s.m/m") "
      Damping constant along direction of
      cable element";
```

```
  parameter SI.ModulusOfElasticity E_bend "
      Modulus of elasticity of bending";
  parameter SI.RotationalDampingConstant
      d_bend "Constant damping constant
      against bending.";
  parameter SI.LinearDensity linearDensity
      "Mass of 1m cable";
  parameter SI.Diameter diameter "Diameter
      of the cable";
  parameter SI.Force f_break "Ultimate
      tensile strength of cable";
end CableData;
```

3.3 Cable Segment

The `CableSegment` model contains the physics of the discrete cable segment. As previously explained, the cable consists of point masses, assembled as cable line via tensile and bending forces. Figure 6 shows the modular approach of the model. The two masses are positioned at the `CableFrame` positions $\underline{r}_a$ and $\underline{r}_b$. Between the two masses, the tensile forces and bending forces are applied. The tensile force is a simple spring/damper force along the normal between the two masses $\underline{e}$, with a nonlinear spring force $f_c(l-l_0)$ and a constant damping parameter d:

$$\underline{f}_{tens,c} = f_c(l-l_0)\cdot\underline{e} \tag{1}$$

$$\underline{f}_{tens,d} = d\frac{\partial}{\partial t}(l-l_0)\cdot\underline{e} \tag{2}$$

The nonlinear spring force $f_c(l-l_0)$ is defined via an interpolated table as forces depending on the lengthening of the cable.

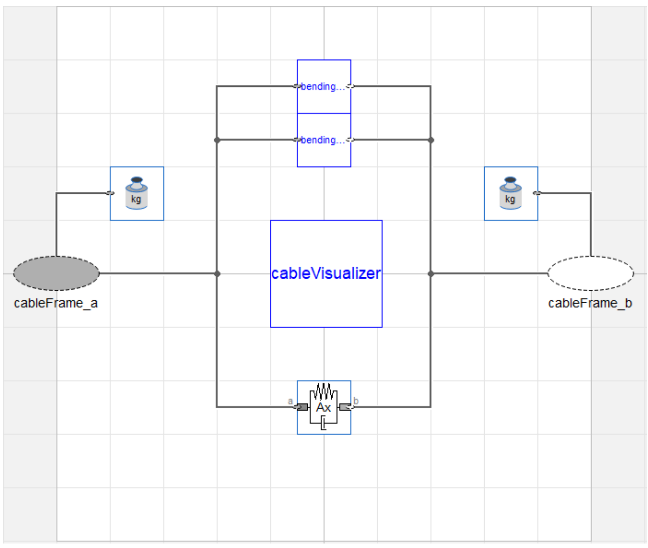

Figure 6. Modelica Implementation of the `CableElement` model

The bending forces can be calculated in two different ways, either with a fast non-physical approximation formula, or via the exact bending radius and modulus of elasticity of the cables material.

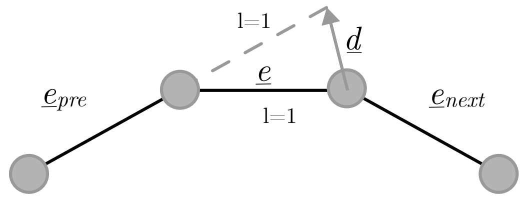

Figure 7. Calculation of the deflection vector $\underline{d}$ between two cable segments. The length of the cable segments is normalized for this calculation

The simplified calculation method to calculate the bending forces uses the deflection vector $\underline{d}$ from the normal axis of the last segment to obtain a measure for the bending of the cable (see Figure 7). The resulting bending forces on the masses at the position $\underline{r}_a$ and $\underline{r}_b$ are then calculate as follows,

$$\underline{f}_{bend,c} = \underline{d}\cdot c_{bend} \tag{3}$$

$$\underline{f}_{bend,d} = \frac{\partial \underline{d}}{\partial t}\cdot d_{bend} \tag{4}$$

$$\underline{f}_{bend} = \underline{f}_{bend,c} + \underline{f}_{bend,d} \tag{5}$$

where c_{bend} and d_{bend} are a non-physical parameter defining the bending stiffness and damping.

The physically correct but slower calculation of the bending forces uses the calculation of the modulus of elasticity of the material: The axes $\underline{e}_{tau,pre}$ and $\underline{e}_{tau,next}$ define the directions of the bending rotation axes between the current segment and the previous, respectively the next section:

$$\underline{e}_{tau,pre} = ||\underline{e}_{pre}\times\underline{e}|| \tag{6}$$

$$\underline{e}_{tau,next} = ||\underline{e}\times\underline{e}_{next}|| \tag{7}$$

Utilizing the angles α_{pre} and α_{next} defining the angle between the current segment and the previous or next segment, the bending radii κ_{pre} and κ_{next} can be calculated as

$$\kappa_{pre} = \frac{\tan(\frac{\alpha_{pre}}{2})}{l_{element}} \tag{8}$$

$$\kappa_{next} = \frac{\tan(\frac{\alpha_{next}}{2})}{l_{element}} \tag{9}$$

The bending torque is replaced with a pair of forces, where the directions are

$$\underline{n}_{pre} = ||(\underline{e}_{tau,pre}\times\underline{e})|| \tag{10}$$

$$\underline{n}_{next} = ||(\underline{e}_{tau,next}\times\underline{e})|| \tag{11}$$

The bending stiffness force on `CableFrame_a` and `CableFrame_b` then results in

$$\underline{f}_{bend,c,a} = -\underline{n}_{pre}\cdot\frac{\kappa_{pre}\cdot E_{bend}\cdot I_y}{l_{element}} \tag{12}$$

$$\underline{f}_{bend,c,b} = \underline{n}_{next}\cdot\frac{\kappa_{next}\cdot E_{bend}\cdot I_y}{l_{element}} \tag{13}$$

 DOI
10.3384/ecp21727

with E_{bend} being the modulus of elasticity of the material and $I_y (= I_x)$ being the inertia of the segment. The bending damping is implemented as

$$\underline{f}_{bend,d,a} = -\underline{n}_{pre} \cdot d_{bend} \cdot \frac{\partial(\kappa_{pre})}{\partial t} / l_{element} \tag{14}$$

$$\underline{f}_{bend,d,b} = \underline{n}_{next} \cdot d_{bend} \cdot \frac{\partial(\kappa_{next})}{\partial t} / l_{element} \tag{15}$$

The summarized bending forces are then obtained with

$$f_{bend,a} = \underline{f}_{bend,c,a} + \underline{f}_{bend,d,a} \tag{16}$$

$$f_{bend,b} = \underline{f}_{bend,c,b} + \underline{f}_{bend,d,b} \tag{17}$$

3.4 Winch

As mentioned in the last section, the winch comprises of a rotating, cylindrical mass with a cable attached to it at a defined point in space. Depending on the type of Winch,

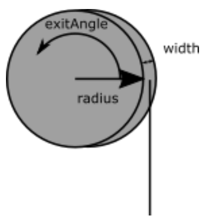

Figure 8. Winch parameters

the position of the cable exit point is either fixed relatively to the non-rotating winch coordinate system (`Winch`) or is moved left and right and with varying radius while the winch rotates (`OscillatingWinch`). In the case of a fixed cable exit point the position defined via the parameter `exitAngle` (α_{exit}, see figure 8):

$$\underline{r}_{lever} = \{r \cdot \cos(\alpha_{exit}), 0, r \cdot \sin(\alpha_{exit})\} \tag{18}$$

For the `OscillatingWinch`, the exit point moves left and right and the radius changes with the rotation angle ϕ of the winch so it is defined as

$$\underline{r}_{lever} = \{r(\phi) \cdot \cos(\alpha_{exit}), y(\phi), r(\phi) \cdot sin(\alpha_{exit})\} \tag{19}$$

where r_{phi} describes a discrete change of the radius ($\pm$ diameter of the cable) when the cable drum is covered with one layer of the cable or one layer has been fully unrolled. The function $y(\phi)$ is described as a saw tooth function to create the meandering movement of the cable with the winch rotation.

The length (and mass) change of the attached cable is defined as function of the rotational velocity $\frac{\partial}{\partial t}\phi$ of the winch:

$$\frac{\partial}{\partial t}\Delta l_{cable} = \frac{r \cdot \frac{\partial}{\partial t}\phi}{\eta_{sf}} \tag{20}$$

Hereby η_{sf} is the so called stretch factor, defining the relative elongation of the cable, potentially resulting from rolling up a cable under tension:

$$\eta_{sf} = l_{cable,current} / l_{cable,nominal} \tag{21}$$

As the stretch factor can change with varying load on the cable, it has to be stored and restored while rolling up or unrolling the cable on the winch. For this purpose, the `spatialDistribution` operator of the Modelica language is being used.
The forces and torques resulting from the cable pulling on the winch are applied also to the MultiBody frame of the winch. For this purpuse, the resulting torques are divided into torques acting on the winch' rotational axis and torques acting in other directions.

3.5 Pulley

A pulley is implemented as two winches, connected on their rotational axes, see Figure 9: The pulley has two po-

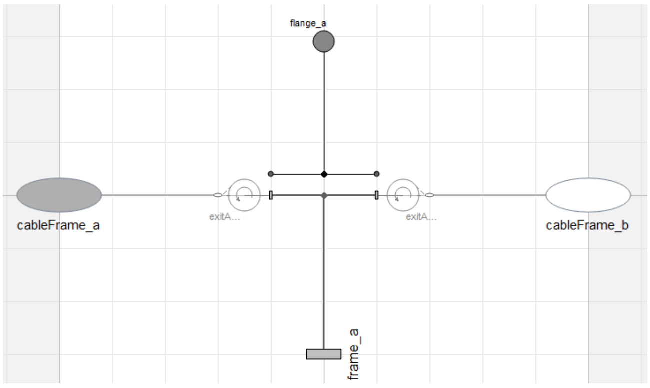

Figure 9. Pulley model

sitions for the cable exit point, defined by the parameters `exitAngleA` and `exitAngleB`.

3.6 Initialization

At the beginning of every simulation run, the spatial position of the cable segments has to be initialized. As there is a multitude of possible initialization scenarios, the initialization routine is implemented as a replaceable function. The base class for this initialization function has the following interfaces:

```
partial function baseCableInitialization
  input Real r_a[3]={0,0,0} "Position of
     cable start";
  input Real r_b[3]={1,0,0} "Position of
     cable end";
  input Real l=10 "Length of cable";
  input Integer n=10 "Number of cable
     elements";
  input Real g[3]={0,0,-9.8} "Gravity";
  output Real r_start[n,3] "start positions
      of cable elements";
  output Real delta_l_init "Length
     difference from initialization";
end baseCableInitialization;
```

The inputs are provided by the cable model as these are known parameters and states. The user can now use this base class to create own initialization routines and calculate the resulting start points of the cable segments `r_start` and a length difference to the nominal cable length `delta_l_init` (e.g. to account for a cable initialized under tension).

There are three initialization routines provided with the library:

1. Catenary initialization:
   Depending on the positions of the ends of the cable and the user-defined length l_{cable}, the catenary line is calculated and all points of the cable segments are initialized on that line:

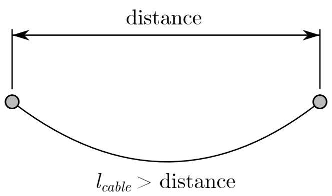

Figure 10. Initialization of a cable on a catenery curve.

2. Linear initialization:
   Depending on the positions of the ends of the cable, the cable segments are distributed equally on the direct connection between those two points, not changing the user-defined cable length l_{cable}:

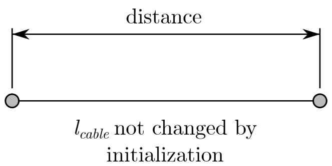

Figure 11. Initialization of a cable on a direct line. As the parameter l_{cable} is not changed by this initialization routine, the cable can be either stretched or compressed.

3. Linear initialization with adaptive length:
   Depending on the positions of the ends of the cable, the cable segments are distributed equally on the direct connection between those two points. The user defined cable length l_{cable} is modified, so that the cable exactly fits between the two mount points. It is possible to set a stretch factor to shorten the cable and to initialize it pre-stretched.

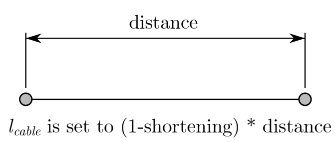

Figure 12. Initialization of a cable on a direct line. The length of the cable is set to distance, but with a defined shortening factor to account for initial tension on the cable.

3.7 Visualization

In order to support a wide range of visualization tools, the library uses replaceable models for the visualizer blocks. Right now, the following visualization tools are supported:

- Modelica tool visualization (Otter, Elmqvist, and Mattsson 2003)
- DLR Visualization library (Hellerer, Bellmann, and Schlegel 2014)
- DLR Visualization2 library (Kümper, Hellerer, and Bellmann 2021)

As cables and winches are easily visualized with cylinders, only few visualizer blocks are necessary.

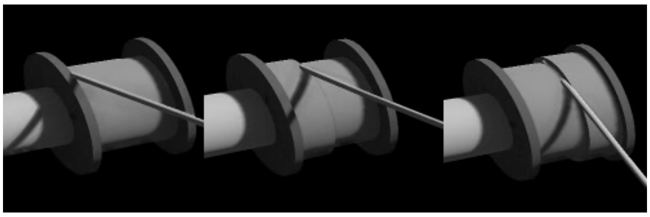

Figure 13. Visualization of the `OscillatingWinch` showing the process of rolling up a cable using the DLR Visualization2 Library.

4 Application examples

In this section, some application examples, combining the DLR Cables Library with Modelica MultiBody commponents are presented. The first example is a construction crane, where the Cables library is used to simulate the mechanics of the load bearing cable and the cable moving the cable trolley. In the second example, the Cables library is used to simulate a gravity compensation mechanism to enable testing of a space robot system on earth.

4.1 Construction Crane

As part of the authors technology transfer activities, a detailed multi-domain model of a construction crane has been developed (see Figure 14).

It is used mainly for teaching and demonstration purposes, and features the mechanical structure, drive trains and cables. The three drive train each consist of the electric drive (asynchronous motor), a vector motor controller, thermal effects, gearbox and brakes (See Figure 15).
For educational purposes, the model also features a calculation of its center of mass, demonstrating the stability of the crane with different loads, varying jib working radius and counterweights.
The model also contains a simplified air drag model for the larger structures and load, further showing the effects of wind load on the crane stability.

The construction crane uses two cable systems and a rotational joint to move its load in three dimensions. The structure of the crane is described in Figure 16. The main

DOI
10.3384/ecp21727

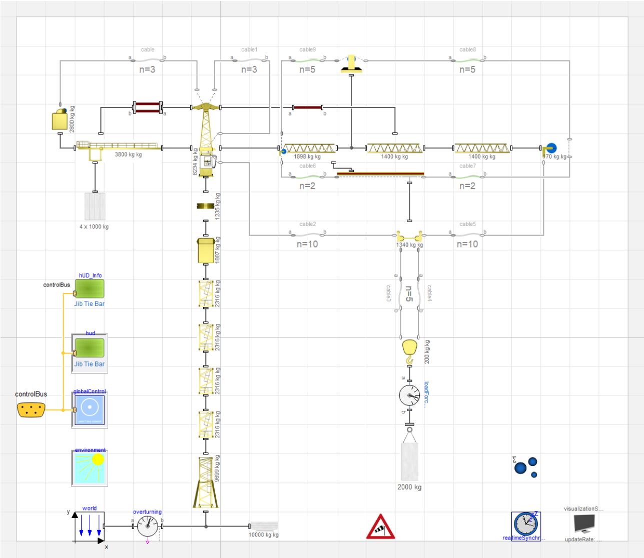

Figure 14. Modelica model of a construction crane. The model can be controlled interactively.

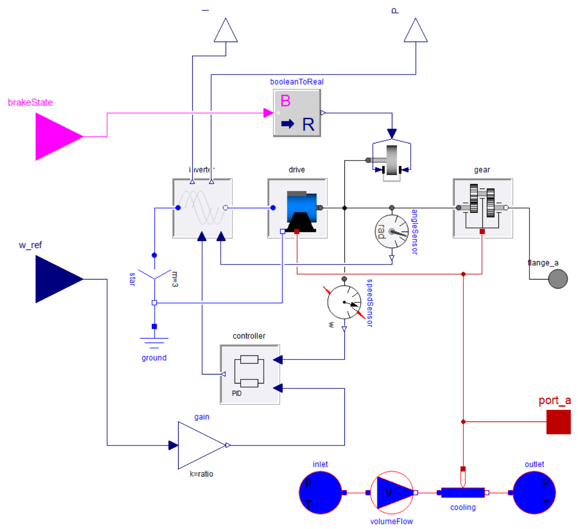

Figure 15. Modelica model of the main winch drive train with asynchronous machine (`IM_SquirrelCage` from the MSL), gearbox, PID controller, inverter, brake and a cooling system.

winch (red) is used to lift the load up and down. The cable attached to it (orange) runs from the main winch in the back of the crane through two pulleys in the tower to the trolley. There, it goes down via another pulley to the load pulley and back up again to another pulley attached to the trolley. It is fixated at the end of the cranes jib. The trolley itself is pulled back and forth with the second cable system (green), featuring a closed loop of three pulleys, each one at the front and end of the crane jib and an actuated one in between. The jib and crane top is rotated with a rotational joint between the crane base and the crane top. In total, the model uses nine cable blocks, one winch and six pulley blocks. With 70 cable segments in total, the model is running in real-time on a Intel Xeon W-2133 with 3.6GHz (Integrator: Sdirk34h). It is visualized in 3D and can be controlled by the user with on-screen controls from the DLR Visualization 2 library (see Figure 17). Figure 18 shows some simulation data of a simple crane maneuver. After initialization, the crane trolley is driven $5m$ forward at $t = 10s$. The top graph shows the position of the trolley, the middle graph the length of the load cables between trolley and load, and the bottom graph the tensile force on the trolley. Several cable related effects are visible, e.g. the swaying of the load, leading to length changes of the two load cables (middle plot), or the forces of the trolley winch system, resulting in the movement of the trolley sled (bottom plot). The crane can be controlled either via an interface to an external PLC, utilizing a ProfiNet interface, or by using on-screen overlays as a graphical joystick and sliders. Additionally, predefined reference trajectories can be used for repeatable simulations.

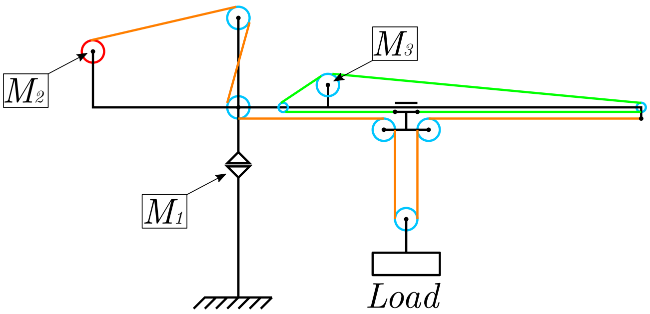

Figure 16. The structure of the construction crane, with the actuated rotational joint (M1), the main winch (M2, red) for the load cable (orange) and the trolley actuator (M3) driving the trolley cable (green). Pulleys are drawn in blue.

Figure 17. Interactive Simulation of the construction crane with on-screen controls (bottom left), real time forces of the jib tie load (middle left), and status information on the drives (upper left).

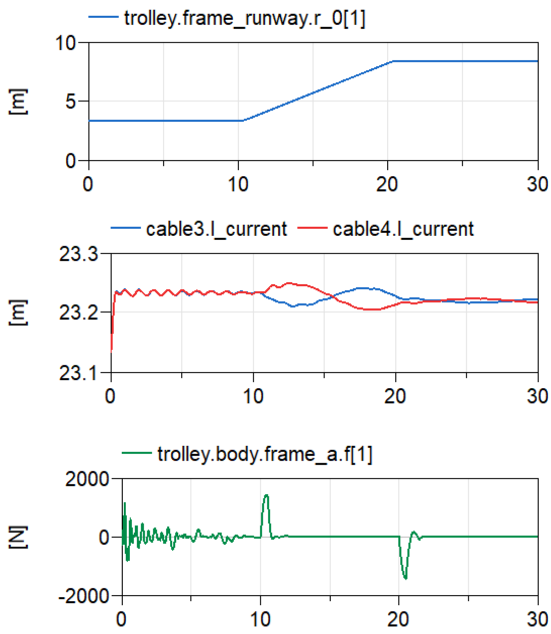

Figure 18. Trolley position (top), length of load cables between trolley to load (middle), combined forces on the trolley (bottom).

4.2 Gravity Compensation for Space Robot Arm

Space robots play an important role in on-orbit servicing operations, including the repair, maintenance and refuelling of satellites. These robotic systems are designed to operate in a zero-gravity environment, thus a gravity compensation mechanism is required to test them on Earth. Therefore, the Motion Suspension System (MSS) (Elhardt et al. 2023), which is visualized in figure 19, has been developed. It allows a force to be applied to the robot in a desired direction. The MSS consists of four cable winches which are arranged in a rectangle on the floor around the robot arm. Pulleys on the ceiling guide the cables to the robot, where they are connected to each other. The suspension cable connects the four cables from the winches to the third segment of the robot arm. The simulation model consists of four pulley blocks, four winch blocks and nine cable blocks with a total of 57 cable segments. The cables are connected to a seven degree of freedom robot using a passive two degree of freedom coupling mechanism. The inputs to the model are the winch torques and the robot torques.

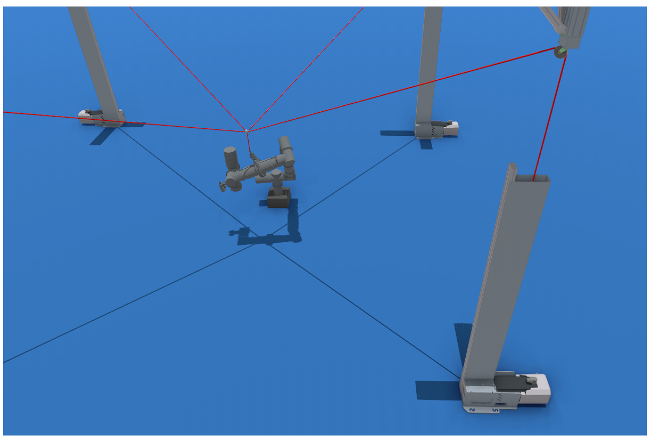

Figure 19. Visualization of the simulated Motion Suspension System.

Figure 20 shows simulation data where the robot is held in a fixed position and the winch torques are increased from an initial 4Nm to 8Nm at t=3s, in order to compensate the gravitational forces acting on the robot. The upper plot shows the forces on the four winch cables and the force on the suspension cable, which connects the four cables to the robot. The lower plot shows the robot joint torques two and three, as these are the joints most supported by the MSS. At the beginning of the simulation, the forces and torques oscillate strongly because the node position of the four winch cables is not perfectly initialised. After about 1.5s the system stabilises and a suspension force of about 130N is applied to the robot. In this case, the drive torques in joints two and three must be about -120Nm and -37Nm respectively for the robot to hold its position. By increasing the winch torque at t=3s, the suspension force is increased to about 260N. In this case the joint torques are reduced to almost zero.

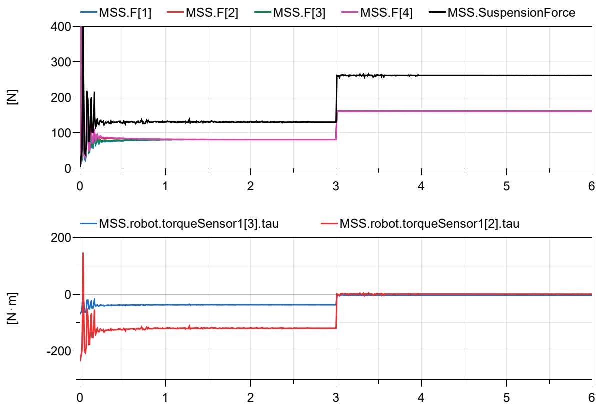

Figure 20. Plots of winch cable forces and suspension cable force (top) and robot joint torques two and three (bottom).

5 Conclusion and Outlook

In this paper, a new pure Modelica based library for fast simulation of cable systems has been presented and its methodical background and implementation has been discussed. An important factor in the reliability of simulation is the thorough validation of the models, in (Seefried and Bellmann n.d.), the validation methods for the cable models, and more real-world examples are discussed. In the future, the library might be extended with more components as more detailed and realistic non-linear bending force models, arbitrary forces acting anywhere on the cable (e.g. to simulate a pulley running along a fixed cable) and collision detection of cables with the environment based on the DLR ContactDynamics Library (Buse, Pignede, and Barthelmes 2023).

DOI
10.3384/ecp21727

References

Berger, Maik and Stefan Heinrich (2011). "Analysis of Belt Drives with circular and variable Pulleys in SimulationX". In: DOI: 10.3403/30219039u.

Börner, Denise (2016-11). URL: https://www.esi-group.com/sites/default/files/news-release/527/pr_esi_group_simulationx_3.8_en.pdf.

Buse, Fabian, Antoine Francois Xavier Pignede, and Stefan Barthelmes (2023-Oktober). "A Modelica Library to Add Contact Dynamics and Terramechanics to Multi-Body Mechanics". In: *Proceedings of the 15th International Modelica Conference*. Vol. 204. Linköping Electronic Conference Proceedings. Modelica Association and Linköping Electronic Conference Proceedings. URL: https://elib.dlr.de/200107/.

Elhardt, Ferdinand et al. (2023-11). "The Motion Suspension System – MSS: A Cable-Driven System for On-Ground Tests of Space Robots". In: *16th International Federation of Theory of Machines and Mechanisms World Congress, IFToMM WC 2023*. Vol. 148. Mechanisms and Machine Science 2. Springer, Cham, pp. 379–388. URL: https://elib.dlr.de/199117/.

Hellerer, Matthias, Tobias Bellmann, and Florian Schlegel (2014-03). "The DLR Visualization Library - Recent development and applications". In: Modellica. ISSN: 1650-3686. DOI: 10.3384/ecp14096899.

Kümper, Sebastian, Matthias Hellerer, and Tobias Bellmann (2021-09). "DLR Visualization 2 Library - Real-Time Graphical Environments for Virtual Commissioning". In: Modelica 2021. ISSN: 1650-3686. DOI: 10.3384/ecp21181197.

Otter, Martin, Hilding Elmqvist, and Sven Erik Mattsson (2003-11). "The New Modelica MultiBody Library". In: *3rd International Modelica Conference*. Ed. by Peter Fritzson. LIDO-Berichtsjahr=2003, pp. 311–330. URL: https://elib.dlr.de/11987/.

Seefried, Andreas and Tobias Bellmann (n.d.). "Validating the DLR Cables Library with Experiments and Parameter Optimization". In: *Asian Modelica Conference 2024*.

Wu, Jianchen et al. (2023-12). "Object-Oriented Modelling of Flexible Cables based on Absolute Nodal Coordinate Formulation". In: Modelica 2023. ISSN: 1650-3686. DOI: 10.3384/ecp20453.