Introducing Dialectic Mechanics

Dirk Zimmer Carsten Oldemeyer

Institute of System Dynamics and Control,
German Aerospace Center (DLR),
{dirk.zimmer, carsten.oldemeyer}@dlr.de

Abstract

This paper introduces a new method for mechanical systems with its own interface that enables the object-oriented formulation of very stiff contacts. It thereby suppresses high frequencies and yields stable replacement dynamics leading to an equivalent steady-state. Potential applications are the efficient modeling and simulation of robotic manipulation or the easier handling of what formerly have been variable-structure systems.

Keywords: multi-body systems, Mechanical contacts and limitations, Robotics

1 Motivation

Libraries for classic multibody simulation have been among the first Modelica libraries ever published. The Modelica Standard Library supports the 3D solution of multibody systems (Otter 2003) with special support for kinematic loops. There are also 1D rotational and translational libraries and a planar mechanical library has been developed that proved its value for teaching purposes (Zimmer 2012) and advanced modeling of gearwheels (van der Linden 2016).

Yet there are modeling tasks that have remained very difficult to master throughout all the years such as:

- The modeling of limited joints
- The modeling of breaking objects
- The modeling of stiction and friction
- The modeling of kinematic loops when reaching maximal extension
- Real-time simulation of hard contacts
- etc.

Our impression is that at least for the Modelica community, progress in these areas has been underwhelming, especially given the high relevance of these issues. For instance, when modeling the manipulation of an object using a robot hand on a robot arm, a combination of any of the above problem may occur.

Many attempts in solving this problem were focused on improving the tooling. Tasks like the modeling of limited joints were identified as variable structure problems or Multi-mode DAEs (Benveniste 2019) and tackled correspondingly by new tools (Mehlhase 2013) or even new languages (Zimmer 2010, Neumayr 2023).

The underlying model equations were practically never questioned. After all, classic Newton mechanics is more than two centuries old (Szabo 1987), and seems hardly worth revisiting.

Au contraire, mon capitan! It is worth revisiting the way we idealize mechanical systems. After all, object-oriented modeling and computers are much younger. We may be able to find a reformulation that enables a better expression of modeler's intent than what was previously conceived. This is the exact aspiration of this paper.

2 On the Idealization of Rigid Body Mechanics

We easily forget that when we model the mechanics of rigid bodies, we model the mechanics of imaginary objects: rigid bodies.

In our real, physical world, there exist no rigid bodies. Everything is elastic and deformable. It is just a matter of degree. If a bullet out of a gun will not convince you, certainly a small piece of space debris as in Figure 1 will:

Figure 1: Impact of a 15g piece of plastic on a block of aluminum with a speed of 24140km/h in public display at NASA Johnson Space Center, Houston, TX, USA.

Rigid bodies thus represent an entirely hypothetical idea, but also a very useful idea. Instead of modeling the pressure waves through an elastic material we can directly formulate non-holonomic constraints and assume an immediate transmission of impulse that upholds the conservation of energy and momentum, since none of these terms can get dissipated in a truly rigid body. We thereby exchange a process that typically operates above 10 kHz (micro-elastic motion within objects) with a process that may often be slower than 10 Hz (macro-motion of objects). Evidently this enables a much more efficient simulation of the kinematic system using far fewer states and much slower eigen-dynamics.

Rigid body mechanics is thus the preferred method to use when we deal with kinematic chains with a fixed number of degrees of freedom. Phenomena as limited joints or stiction can consequently be interpreted as varying the number of degrees of freedom. When regarding such problems as discrete configuration changes, this leads straight to the previously mentioned approaches (Zimmer 2010, Mehlhase 2013, Benveniste 2019, Neumayr 2023) for variable structure systems. Also discrete Dirac impulses then need to be considered as in (Zimmer 2006).

However, even if a (potentially very complex) solution for discrete configuration changes is available, it is often inappropriate to apply since it forces us to simplify by discretization the very thing we actually want to focus on. Whether a gripping mechanism is actually holding an object or not and when and up to what degree is a question that is not easily answered by yes or no. When going into detail, one may detect many transient states.

For such cases, the modeler is now forced to re-establish elastic bodies at least for the region of contact dynamics. Whereas he may succeed, to keep the set of state variables small, applying realistic constants for the elasticity will often yield high frequency behavior or other ill-suited eigen-dynamics that drastically lower the simulation efficiency. This is especially true when a stiff object is tightly gripped, and notably it is the very intent of gripping devices to grip things tightly in order to create a force-locked connection.

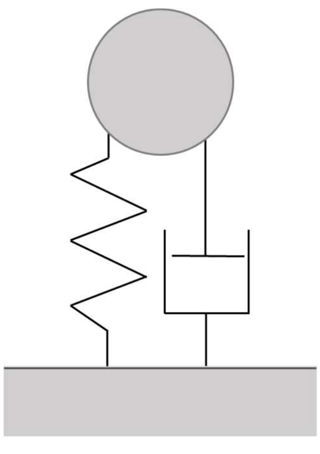

Figure 2: A one-dimensional spring-damper system modeling an elastic contact with ground.

For illustration, let us look at the simple 1D mechanics of a spring-damper system as in Figure 2.

$$v = \frac{ds}{dt} \tag{1a}$$

$$\frac{dv}{dt} = \frac{f}{m} + g \tag{1b}$$

$$f = -cs - dv \tag{1c}$$

where s is the position and v is velocity. The force f results out of the spring damper dynamics with their respective coefficients c and d . g is the gravity acceleration.

For $d > 0$ and $m > 0$, this system reaches a steady-state solution at:

$$s = \frac{mg}{c}; v = 0$$

The eigenvalues of the system are well known:

$$\lambda_{1,2} = -\frac{d}{2m} \pm \sqrt{\frac{d^2}{4m} - \frac{c}{m}}$$

Let us suppose, we as modelers are willing to sacrifice the precision of the transient dynamics for the sake of simulation efficiency. Since both m and c contribute to the steady-state solution, we may hence only modify the damping constant d.

Below critical damping we may move the eigenvalues only alongside a circle in the plane of imaginary numbers. This helps at least avoiding high frequencies and is often feasible for implicit ODE solvers. Going beyond critical damping makes matters even worse, causing one eigenvalue to become highly negative whereas the other starts to interfere with potentially other slow dynamics that may exist in extension of this system. The direct manipulation of d in a complex system is often cumbersome because a favorable choice depends on the values for spring constants and masses for the configuration.

Despite its tight limitations, this method is often applied and for real-time simulation, many simulation practitioners are desperate enough to even manipulate constants for masses or springs (Neves 2019, Reiser 2021), often leading to a virtual world of strangely wobbling objects.

3 The Idea of Dialectic Mechanics

When practitioners show such signs of desperation, it is mostly because their model does not match their original intent.

Indeed, it is not very intuitive for us why the gripping of an object is such a tough task to simulate, our brain simulates it all the time and it seems to do a pretty good job at it despite being a low-frequency computational device (albeit being massively parallel). We thereby intuitively decompose the macroscopic motion of our arm, hand, and object from the microscopic motion of the object in the tension-regime of the gripping hand. The first motion is dominated by the kinetic forces resulting from the acceleration of objects, the latter motion is dominated by the elastic forces resulting from the positional shift of the object.

Realizing such a decomposition in form of equations is unfortunately not intuitive at all but it can be achieved:

- We denote the velocity in the elastic regime: v_{el}
- We denote the velocity in the kinetic regime: v_{ki}

In an ideal world $v_{el} = v_{ki}$. However, to express the modeler's intent of splitting into two regimes, we formulate:

$$\frac{dv_{ki}}{dt} T_D = v_{el} - v_{ki} \tag{2a}$$

with T_D being denoted as dialectic time-constant. This represents a first-order filter for the kinetic motion. High-frequency motion in the elastic regime are therefore inhibited for their impact on the kinetic regime. Let us now restate the equations of our spring damper system:

We can compute the elastic force f_{el}:

$$f_{el} = -cs + mg \tag{2b}$$

With

$$\frac{ds}{dt} = v_{el} \tag{2c}$$

We can compute the kinetic force f_{ki}:

$$f_{ki} = -m\frac{dv_{ki}}{dt} - d \cdot v_{ki} \tag{2d}$$

Evidently, the decomposition of velocity led us to decompose also the forces and we now treat elastics and kinetics as separate phenomena. In order to rejoin them to a consistent solution, we remember our equation (1) of the first-order filter and enforce the balance of forces:

$$f_{el} + f_{ki} = 0 \tag{2e}$$

This is why we call this approach: dialectic mechanics. If we personify the phenomena of elastics and kinetics then both persons would argue for their regime by expressing their respective force. In the end, they have to reach a common conclusion that neutralizes their respective counterarguments.

In correspondence, this system of equations has two states: the position s belonging to the elastic domain and v_{ki}, belonging to the kinetic regime. We can plug in Equation (2b) and (2d) in Equation (2e) to eliminate the forces:

$$\frac{dv_{ki}}{dt} = g - \frac{c}{m}s - \frac{d}{m}v_{ki} \tag{3a}$$

and plugging in Equation (2a) in (2c) eliminates v_{el}:

$$\frac{ds}{dt} = gT_C - \frac{cT_D}{m}s + \left(1 - \frac{dT_D}{m}\right)v_{ki} \tag{3b}$$

We see that for $T_D \rightarrow 0$ this system becomes equivalent to the original system of Equations (1a-1c). Small values for T_D shall thus result in a small deviation. We also see that T_D has no impact on the steady-state solution, which is still:

$$s = \frac{mg}{C}; v_{ki} = 0$$

But the eigen-dynamics are now manipulated so that we have new eigenvalues:

$$\lambda_{1,2} = -\frac{d + cT_D}{2m} \pm \sqrt{\frac{(d + cT_D)^2}{4m^2} - \frac{c}{m}} \tag{4}$$

The term in the square root is now a quadratic function on c/m with a minimum at:

$$\left(\frac{c}{m}\right)_{min} = \frac{2}{T_D^2} - \frac{d}{mT_D}$$

and the minimum value of:

$$-\frac{1}{T_D^2} + \frac{d}{mT_D}$$

For an undamped system with $d = 0$, this simplifies to:

$$-\frac{1}{T_D^2}$$

which limits the imaginary part of the eigenvalues to not exceed $\pm iT_C^{-1}$, corresponding to a maximum rotation of

$$\omega_{max} = T_D^{-1}$$

or a frequency limitation of $\frac{1}{2\pi T_D}$. In the original undamped system, the angular velocity was simply:

$$\omega_S^2 = \frac{c}{m}$$

The dialectic undamped system yields a different rotation:

$$\omega_D^2 = \frac{c}{m} - \left(\frac{c}{m}\frac{T_D}{2}\right)^2$$

which (for $\omega_D > 0$) can be expressed in terms of ω_S:

$$\omega_D^2 = \omega_S^2 \left(1 - \omega_S^2 \frac{T_D^2}{4}\right)$$

We see that the deviation from the original system is small for low frequencies but keeps rising quadratically up to and beyond the frequency limitation. From equation (4) we can also see that there is an additional damping term added with the strength of $\omega_S^2 T_D/2$.

In terms of eigenvalue manipulation: what is subtracted on the imaginary axes is added on the left side of the real axis (for an undamped system). This means that our error is of stabilizing (or dissipative) nature. Indeed, we can see from Equation (4) that working with T_D is equivalent to manipulating the damping constant. The time-constant however offers a systematic approach to perform this: eigenvalues near the center are only little influenced, eigenvalues close to the frequency limitations are drastically manipulated. Also, we still have the original damping coefficient d available for further manipulation of the eigenvalues in case this is needed.

If the slow dynamics of interest is well below the imposed frequency limitation, we can expect our error to be within an acceptable range for many practical applications, especially those applications where the model uncertainty is quite high like gripping little known objects. We shall also remember that the steady-state solution is not manipulated.

4 Object-Oriented Formulation

4.1 1D Translational Systems

All what has been discussed in the previous section has just been the eigenvalue manipulation of a small system with two states. This would not deserve our attention, if the conclusion remains restricted to this problem class. Fortunately, the idea of dialectic mechanics is very well suited for an object-oriented formulation, which allows its application to larger and more complex kinetic constructs.

To this end, let us review the core idea and devise a 1D library for translational mechanics. The first key idea was to split the mechanics into two regimes:

- The elastic regime, taking care about position and storage of potential energy such as springs or gravity.
- The kinetic regime, taking care about dissipation and storage of kinetic energy

We can represent these two regimes, by two corresponding pairs of effort and flow:

Listing 1. 1D-connector implementation

```
connector Flange
  SI.Position s;
  flow SI.Force f_el;

  SI.Velocity v;
  flow SI.Force f_ki;
end Flange;
```

We also define that $v_{el} := \frac{ds}{dt}$ and if not stated explicitly otherwise $v_{ki} := v$ and the acceleration is $a = dv_{ki}/dt$. When we implement the components, we simply do so in a dialectic manner. We set up the equations for each of the regimes independently.

The fixation is boring as usual:

Listing 2. Component for a fixed position

```
model Fixed
  Interfaces.Flange_b flange_b;
  parameter SI.Position s;

equation
  flange_b.s = s;
  flange_b.v = 0;
end Fixed;
```

The element for translation now has to contain the derivative of the non-holonomic constraint in the kinetic domain. Kinetic and elastic forces are independently transferred.

Here is the implementation of a body component:

Listing 3. Component representing a 1D mass

```
model Body
  Interfaces.Flange_a flange_a;

  parameter SI.Mass m;
  parameter SI.Acceleration g = -9.81;

  SI.Acceleration a;
  SI.Velocity v(stateSelect= ...avoid);
  SI.Position s(stateSelect= ...avoid);

equation
  a = der(v);
  s = flange_a.s;
  v = flange_a.v;

  flange_a.f_ki  = m*a;
  flange_a.f_el  = -m*g;
end Body;
```

Please note that the gravity is attributed to the elastic domain since it represents a potential force depending on position (albeit not in this particular example). Also, the body component does not state that the velocity is derivative of the position. Other than a typical body component, it does not define states.

To finally join the two regimes and reach a common conclusion, we have to apply the filter equation that relates v_{el} and v_{ki} and also enforce the balance of forces: $f_{el} + f_{ki} = 0$. This has to happen where we define our degrees of freedom for the motion of the system. These are the joint elements. In 1D mechanics there is only 1 degree of freedom and hence only one type of joint: the prismatic joint.

Listing 4. The prismatic joint in 1D

```
model Joint
  Interfaces.Flange_a flange_a;
  Interfaces.Flange_b flange_b;
  RealInput f_ext;
  parameter SI.Time TD;
  SI.Position s(stateSelect = …prefer);
  SI.Velocity v(stateSelect = …prefer);
  SI.Velocity v_el(start = 0);

equation
  flange_a.s + s = flange_b.s;
  flange_a.f_el + flange_b.f_el = 0;
  flange_a.v + v = flange_b.v;
  flange_a.f_kin + flange_b.f_kin = 0;
  flange_a.f_el + flange_a.f_kin =f_ext;
  v_el = der(s);
  der(v)*TD = (v_el - v);
end Joint;
```

In dialectic mechanics, typically s and v_{ki} are chosen as states of the system. A linear system has then to be solved, in order to solve for v_{el} with the balance of forces $f_{el} + f_{ki}$ forming the corresponding residual. In this particular component model, this sum adds up not to zero but to an external force f_{ext} that can be used to actuate the joint.

Following the same spirit, we can model an asymmetric spring-damper to model a mechanical stop element.

Listing 5. ElastoGap model

```
model ElastoGap
  Interfaces.Flange_a flange_a;
  Interfaces.Flange_b flange_b;
  parameter SI.Position l;
  parameter SI…SpringConst. c;
  parameter SI…DampingConstant d;
  SI.Position ds( start = 0);
  SI.Velocity dv( start = 0);

equation
  flange_a.s + l + ds = flange_b.s;
  flange_a.f_el + flange_b.f_el = 0;
  flange_b.f_el = if ds < 0 then ds*c
                  else 0;
  flange_a.v + dv = flange_b.v;
  flange_a.f_kin + flange_b.f_kin = 0;
  flange_b.f_kin = if ds < 0 then dv*d
                   else 0;
end ElastoGap;
```

The following model uses two of such elasto-gaps to model a 500g ball clamped into two pieces of hard wood with an indentation of 0.1mm resulting in a spring constant of roughly 2MN/m. The system is modelled without any damping (which is totally unrealistic). The whole construction is then moved by two subsequent and counteracting force impulses. The corresponding setup of Figure 3 can be regarded as a very simplistic model of a robotic grip holding and moving an object.

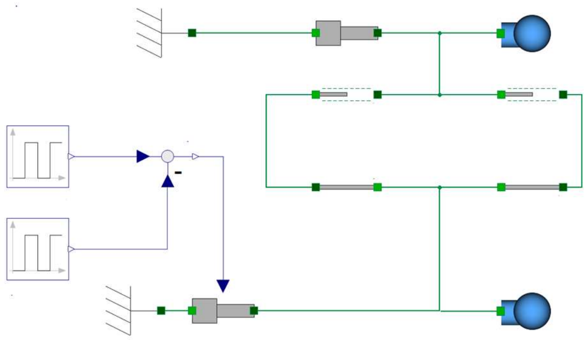

Figure 3: Modelica Diagram of a clamp on a fixed actuator. The upper body is squeezed between to elasto-gap models. The lower body represents the cartridge that is being moved by two force impulses.

The simulation plot in ***Figure 1***Figure 4 below shows the result of the corresponding simulation using two different time constants 1 microsecond and 1 millisecond. The system has been simulated in both cases with Runge-Kutta of 3rd order, using the corresponding step-width.

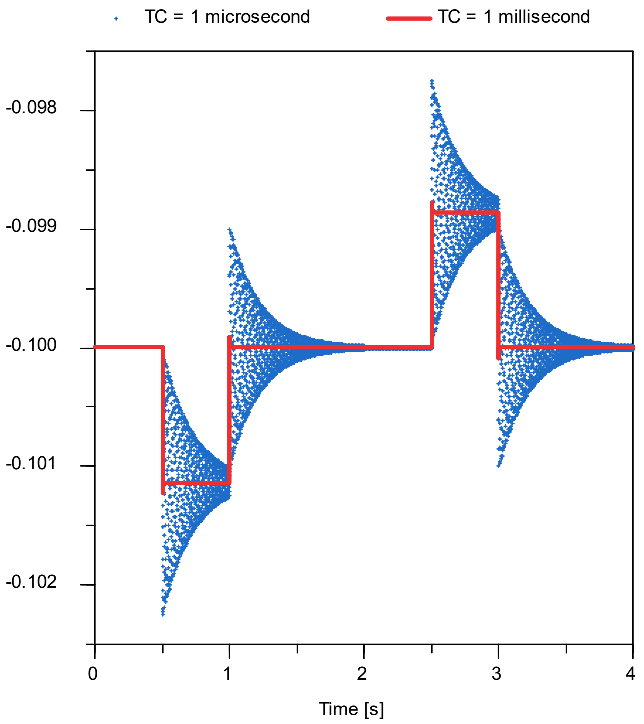

Figure 4: Penetration depth [mm] into the left clamp component represented by an elasto-gap, for the choice of two different time constants (TC = T_D). Both agree on the time-averaged solution.

Using a microsecond as time constant, we can see the resulting high-frequency solution in the contact region of the idealized hard-wood. Ideally, the oscillation should last forever (since no damping is assumed) but the small added damping lets the oscillation decay roughly within a second.

Using a millisecond as time constant, the system is almost perfectly damped artificially but exhibits the same shift in its quasi-equilibrium. The minute changes in penetration depth due the acceleration of the body are correctly assessed (on time-average basis).

4.2 1D Rotational Systems

Using strict analogy, a 1D-rotational library can be created. Here we use two angular velocities: ω_{el} and ω_{ki} to establish the dialectic regimes where again a balance of torque $\tau_{el} + \tau_{ki}$ forms the root of the equation system.

5 Complex Kinematics

To demonstrate the suitability of dialectic mechanics for complex kinematics, we have developed a planar mechanical library, similar to (Zimmer 2012).

As connector we use 2x3 pairs of potential and flow variables.

Listing 6. Planar mechanical connector

```
connector Frame
  //elastic regime
  SI.Position x;
  SI.Position y;
  SI.Angle phi;
  flow SI.Force fx_el;
  flow SI.Force fy_el;
  flow SI.Torque t_el;

  //kinetic regime
  SI.Velocity vx;
  SI.Velocity vy;
  SI.AngularVelocity w;
  flow SI.Force fx_ki;
  flow SI.Force fy_ki;
  flow SI.Torque t_ki;
end Frame;
```

The implementation is in strong correspondence, with the 1D translational library. For the sake of brevity, the code of the prismatic joint, is to be regarded as exemplary and provides sufficient insight into the general dialectic modeling style:

Listing 7. A prismatic joint in a planar world

```
model Prismatic "A prismatic joint"
  extends DialecticPlanarMechanics.Inter
faces.PartialTwoFrames;

  parameter Boolean useFlange=false;
  parameter SI.Time TD;
  parameter SI.Position r[2]
  final parameter SI.Length l=sqrt(r*r);
  final parameter SI.Distance e[2]= r/l

  Translational1D…Flange_a flange_a(
    s=s,v=v,
    f_el=f_el,f_kin=f_kin) if useFlange;

  SI.Position s(stateSelect = …prefer);
  SI.Velocity v(stateSelect = …prefer);
  SI.Velocity v_el;
  SI.Force f_el;
  SI.Force f_kin ;
  Real e0[2] ;
  SI.Position r0[2];
  Real R[2,2];

equation
  R={{cos(frame_a.phi),-sin(frame_a.phi)},
     {sin(frame_a.phi),cos(frame_a.phi)}};
  e0 = R*e;
  r0 = e0*s;

  //elastic regime
  frame_a.x + r0[1] = frame_b.x;
  frame_a.y + r0[2] = frame_b.y;
  frame_a.phi = frame_b.phi;
  frame_a.fx_el + frame_b.fx_el = 0;
  frame_a.fy_el + frame_b.fy_el = 0;
  frame_a.t_el  + frame_b.t_el
  + r0*{frame_b.fy_el,-frame_b.fx_el}
  = 0;

  //kinetic regime
  frame_a.vx - r0[2]*frame_a.w + v*e0[1]
  = frame_b.vx;
  frame_a.vy + r0[1]*frame_a.w + v*e0[2]
  = frame_b.vy;
  frame_a.w = frame_b.w;
  frame_a.fx_kin + frame_b.fx_kin = 0;
  frame_a.fy_kin + frame_b.fy_kin = 0;
  frame_a.t_kin  + frame_b.t_kin
  + r0*{frame_b.fy_kin,-frame_b.fx_kin}
  = 0;

  //synergy
  v_el= der(s);
  der(v)*TD = (v_el- v);
  {frame_b.fx_el,frame_b.fy_el}*e0
   + {frame_b.fx_kin,frame_b.fy_kin}*e0
   + f_el + f_kin = 0;


  //actuation force
  if not useFlange then
    f_el = 0;
    f_kin = 0;
  end if;

end Prismatic;
```

Please note also, that the prismatic joint contains 1D-flange for actuation. Combing this joint with the 1D-elasto gap model provides for instance the opportunity to model limited joints in a natural way, without needing any extra components.

Indeed, when we combine this prismatic joint with a 1D-Elasto Gap model, we get a limited prismatic joint. We can then use then this joint to create the simple model of a thread pendulum as presented in Figure 5.

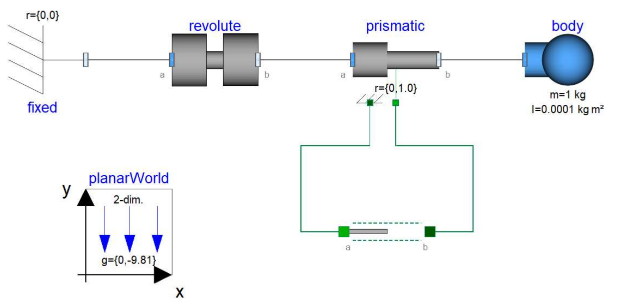

Figure 5: Model of thread pendulum. An elasto-gap is used to model the maximum extension of the thread.

Under the chosen initial conditions, the pendulum first swings through the lower hemicircle before it reaches its apogee and entering free fall conditions as in Figure 6. From then on, it sharply drops into its own thread, continually bouncing off the confining circle of the pendulum. This is because the thread is quite stiff with a spring constant of 1MN/m but only lightly damped with a damping constant of 1kNs/m

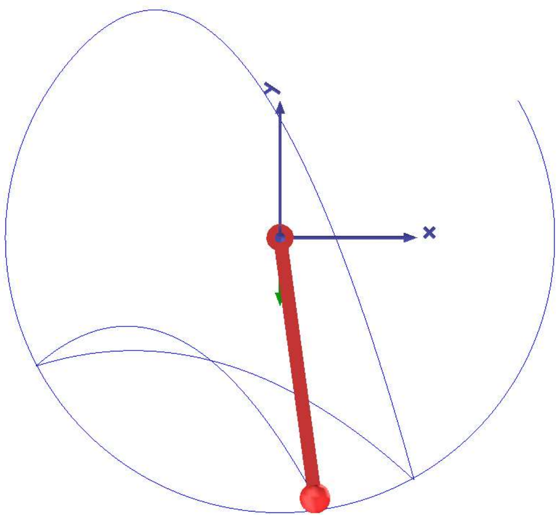

Figure 6: Trajectory of the thread pendulum for the first 2.6 seconds.

The simulation thus exhibits both slow mode and fast mode behavior. The first second with its swing through the hemicircle represents a slow mode behavior. Independent of the time constant for T_D all simulations agree on the elongation length of the thread due to the centrifugal and gravitational forces acting on the mass. There is only a slight phase shift depending on T_C visible in Figure 7.

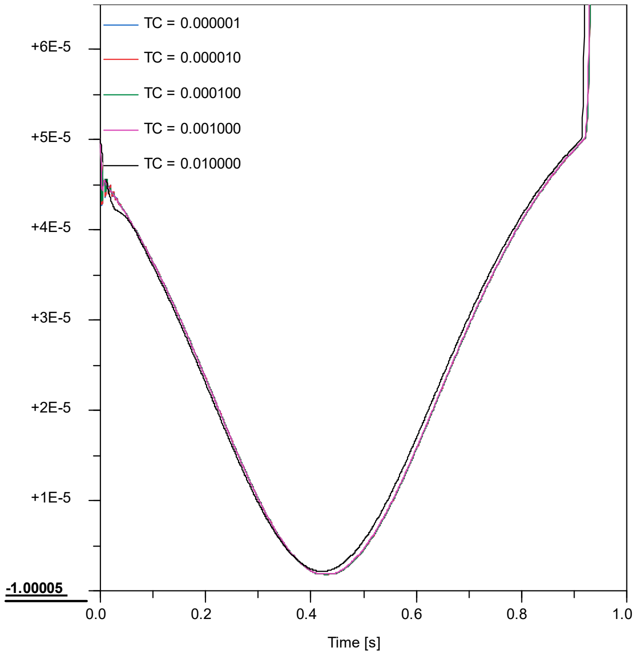

Figure 7: Extension of thread in meter through the first hemicycle due to gravity and centrifugal forces. In the slow-mode, the agreement of models with different time constants (TC = T_D) is high.

The bounce off its own thread represents a fast mode behavior. Here significant differences become visible in Figure 8 with respect to the choice of T_D. Low values for T_D lead to an artificially dampened system that dissipates its energy much quicker. This is exactly what is expected from the previous eigenvalue analysis.

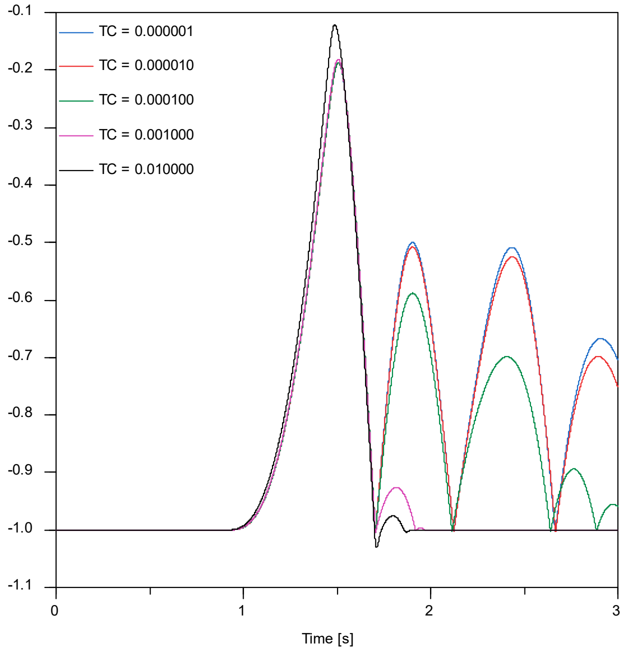

Figure 8: Center distance of pendulum body in meter during the overall trajectory. The artificial dampening with increased time constants (TC = T_D) impacts the elasticity of the bounce.

5.1 Kinematic Loops

When using dialectic mechanics, all joints always express state variables. Kinematic loops are closed using an elastic element (which however can be very stiff). The system therefore has more states than the classic set-up of kinematic loops, but avoids the formulation of a non-linear equation system. To solve for the balance of forces, still only a linear system of equations needs to be solved. Figure 9 presents is a simple 2D-kinematic loop for the extension of a landing gear.

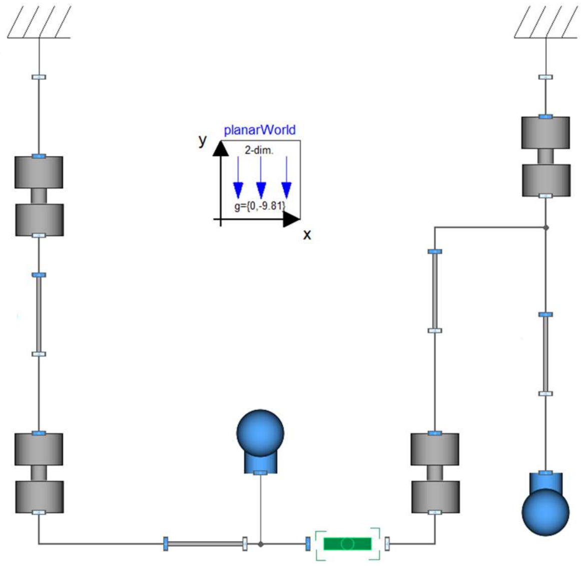

Figure 9: Model diagram of a simple unfolding kinematic of a landing gear. The loop is closed by the green component representing a spring-damper element (translational and rotational) with high stiffness.

This example has 8 state variables (the angles and the kinetic angular velocities of the revolute joints) and there is one linear implicit equation system of size that can be torn by 4 iteration variables: the 4 elastic angular velocities

One advantage of using an elastic element for closing loops is that typical singular points of maximal extension can now be properly handled. A fully rigid formulation exhibits a singular point at its point of maximum extension as depicted in Figure 10 because the kinetic energy at this point has nowhere to go. Using the elastic element for loop closure avoids this problem and the elastic elements can take the impulse from the kinematic reaching its limits.

As this example shows, even impulses on kinematic loops can be handled by dialectic mechanics. Because of the suppression of high frequencies, stiff springs can be used for closing kinematic loops without creating high frequencies. As with the example of the thread pendulum, the applied time-constant matters for the fast-mode behavior of the impulse but not for the slow unfolding dynamics.

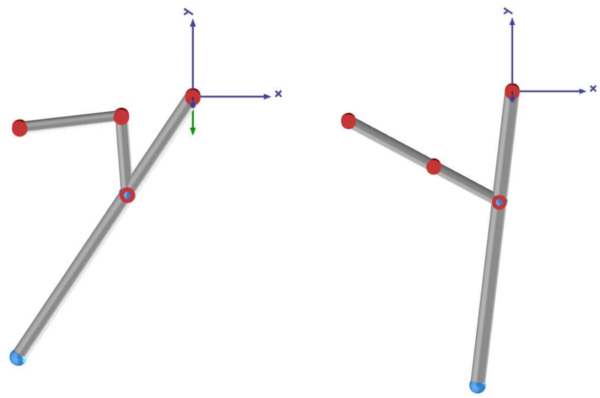

Figure 10: Visualization of the kinematic loop in two different states: unfolded on the right and partially folded on the left.

6 Conclusions

Let us now recapitulate on what we have actually implemented. Usually for a mechanical library, one of the first equations to write down would be:

$$a = \frac{d^2 s}{dt^2}$$

The acceleration is the second time-derivative of the position. What else should it be? Remarkably, this equation is not fulfilled in dialectic mechanics. Here we only make an approximation for lower frequencies.

$$a \approx \frac{d^2 s}{dt^2}$$

Effective modeling always represents an effective (and thereby lossy) compression of reality. It is hence all about doing an error on purpose where it is the most helpful. First of all, the rigid body assumption only holds up for low frequencies. At high frequency excitation, all bodies increasingly appear to be elastic. Limiting the frequency bandwidth is hence simply a consequential alignment to the rigid body assumption.

By making the acceleration only an approximation of the second derivative of position, we enable a disassociation of the regime for kinetic energy from the regime of potential energy. This disassociation enables the direct transfer of energy within these domains, especially within elastic elements.

Figure 11 illustrates a light-weighted, weakly damped body in the center of two very stiff springs and dampers.

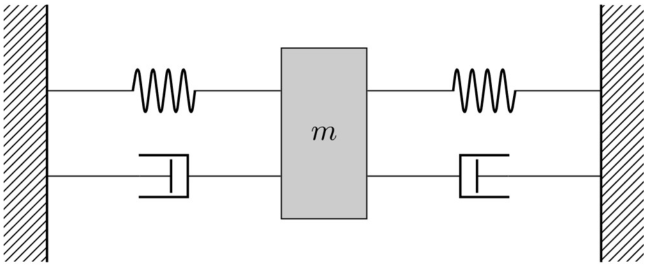

Figure 11: Classic representation of a body clamped in by two spring-damper elements. Any change of the potential energy stored in the springs has to go through the kinetic energy.

In the classic formulation any transfer of potential energy between the two springs has to go through the kinetic energy of the body component. This enforces very high frequencies.

In dialectic mechanics, we split the two regimes and couple them by a low-pass filter. This is illustrated in Figure 12.

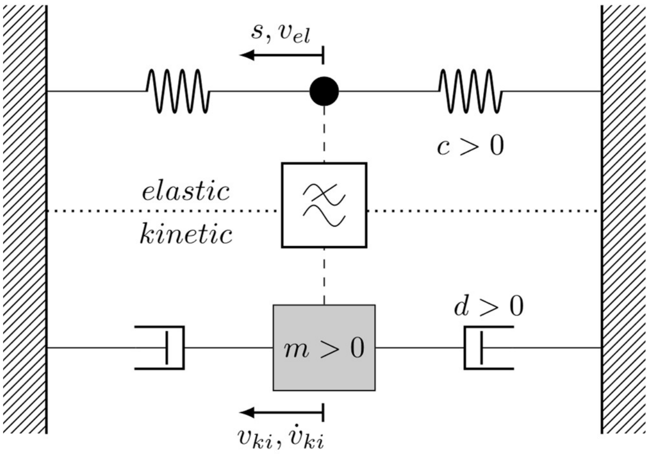

Figure 12: Dialectic view of the same system. On the elastic side, the body is now represented by a massless-point where only its low-frequency motion passes through the mass-holding body on the kinetic side. This enables an independent energy exchange of the potential energy stored in the springs. In this way, the elasto-static equilibrium can be found without transferring all energy through the body, instead it is (to a various degree) dissipated in the filter.

The dot connecting the two springs is now massless and only connected to the original mass by the first-order low-pass filter. In this way, a direct (dissipative) energy transfer between the springs is enabled and also the filter equation furthermore ensures that the springs are always undergoing a continues motion. Hence, the steady-state solution can be reliably and consistently found while avoiding higher frequencies.

This is useful because a lot of mechanical phenomena can be quite conveniently modelled using very stiff springs:

- Limited joints
- Contact dynamics
- Stiction
- etc…

The reason modelers learn to avoid very stiff springs is that they typically yield highly unfavorable eigendynamics. Using dialectic mechanics, the modeler can now use realistic spring constants without a bad conscience for many applications since the manipulation of eigenvalues for high frequencies keeps the dynamics in check without modifying the steady-state solution and only causing small errors for the dynamics of the slow modes. This greatly eases modeling of all of the above phenomena.

Regarding impulses: dialectic mechanics works fine for inelastic contacts. Fortunately, many gripping mechanisms are designed to provoke inelastic contacts having multiple layers of material with high damping constants (like human hands). For purely elastic contacts, there is a substantial error and the conservation of energy and momentum is disregarded. The error gets worse, the harder the material. With being too dissipative, the error is at least benevolent, meaning that it does not destabilize the system and enables a robust solution nonetheless.

The robustness and the avoidance of non-linear equation system in implicit form makes dialectic mechanics especially suited for the hard real-time simulation using explicit solvers. To this end, the presented manipulation of eigenvalues is however not sufficient and a further manipulation needs to be applied. Together with an extensive error analysis these are presented in the corresponding follow-up paper (Oldemeyer 2023). Interestingly also other approaches for explicit solvers split the time-domain and use a two-fold model approach such as (Peiret 2020).

Two remarks regarding the interface of dialectic mechanics. First remark: it is of course possible to model the ideal case where $v_{el} = v_{ki}$ using this interface as well. The interface would then be partly redundant which in consequence simply yields a slightly bloated formulation of classic multibody mechanics. In principal, mixing of approaches is hence possible. For the example of on-orbit servicing of satellites, the satellite trajectories could be modeled with ideal equations ensuring the conservation of momentum in space. The robotic interaction between satellites could then be modeled using a dialectic approach.

Second remark: Dialectic Mechanics is part of a larger modeling class denoted as Linear Implicit Equilibrium Dynamics (Zimmer, 2023). This class of models has originally been conceived to enable robust modeling of thermofluid systems but it revealed application potential outside this domain as well. Linear Implicit Equilibrium Dynamics is also a class of models whose compilation scheme is comparable simple and enables a generation of simulation code per component. This could be useful for mechanical libraries in a more dynamics run-time setting.

References

Benveniste, A., Caillaud, B., Elmqvist, H., Ghorbal, K., Otter, M., Pouzet, M. (2019). "Multi-Mode DAE Models - Challenges, Theory and Implementation". In: *Computing and Software Science. Lecture Notes in Computer Science*, vol 10000. Springer, Cham. https://doi.org/10.1007/978-3-319-91908-9_16

van der Linden, Franciscus L. J. (2016) *Gear contact modeling for system simulations and experimental investigation of gear contacts*. Dissertation, Technische Universität München.

Mehlhase, A. (2013) A Python framework to create and simulate models with variable structure in common simulation environments. *Mathematical and Computer Modelling of Dynamical Systems*, Vol. 20(6), pp. 566—583

Neumayr, Andrea & Otter, Martin. (2023). "Modelling and Simulation of Physical Systems with Dynamically Changing Degrees of Freedom." *Electronics*. 12. 500. 10.3390/electronics12030500.

Neves, Miguel (2019) *Human-In-The-Loop Controlled Lunar Landing Simulator*. Master Thesis, Technical University of Munich.

Oldemeyer, C., D. Zimmer (2023), "Hard Real-Time Simulation using Dialectic Mechanics", *Proceedings of 15th Modelica Conference,* Aachen, Germany

Otter, M., H. Elmqvist and S.E. Mattsson (2003), "The New Modelica MultiBody Library," *Proc. 3rd International Modelica Conference*, Linköping, Sweden, pp.311-330.

Peiret, A., González, F., Kövecses, J., and Teichmann, M. (2020). "Co-Simulation of Multibody Systems With Contact Using Reduced Interface Models." ASME. J. Computational. Nonlinear Dynamics. 15(4):

Reiser, Robert (2021) Object Manipulation and Assembly in Modelica. In: *Proceedings of 14th Modelica Conference 2021, Linköping, Sweden*, pp 433-441. doi: 10.3384/ecp21181433.

Istvan Szabo (1986): *Geschichte der mechanischen Prinzipien*, 3. Auflage. Birkhäuser Verlag Basel.

Zimmer, D. (2012), A Planar Mechanical Library for Teaching Modelica *Proceedings of the 9th International Modelica Conference* , Munich, Germany

Zimmer, D. (2010), *Equation-Based Modeling of Variable Structure Systems*, PhD Dissertation, ETH Zürich, 219 pages

Zimmer, D. (2023), "Object-Oriented Formulation and Simulation of Models using Linear Implicit Equilibrium Dynamics" *Proceedings of 15th Modelica Conference,* Aachen, Germany