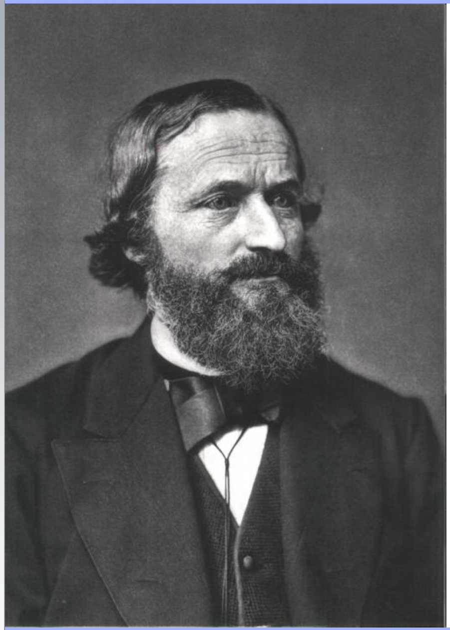
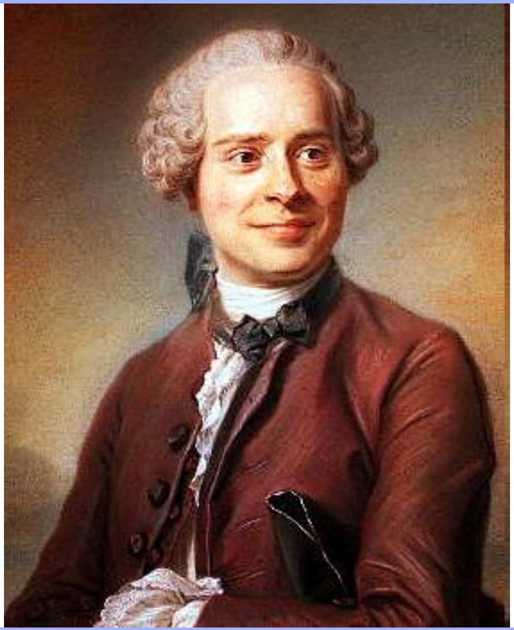
Virtual Physics
Equation-Based Modeling

TUM, November 08, 2022

Object-oriented formulation of physical systems – Part I





Dr. Dirk Zimmer

German Aerospace Center (DLR), Robotics and Mechatronics Centre

Object-Oriented Languages

- One of the first programming languages that was designed for the main purpose of general computer simulation was Simula 67.

- It was designed in the 1960s, and it is also known to be the first object-oriented language in programming language history.

- Whereas many concepts and design ideas of Simula have been quickly adopted by many mainstream programming languages like C++, JAVA, or Eiffel, the development of equation-based object-oriented modeling languages took unfortunately much longer.

- In spite of common origins, this led partly to a dissociation of the corresponding object-oriented terminologies. Object-orientation in programming languages is thus partly distinct from its representation in the equation-based counterparts.

Object-Orientation in Physics

Robotics and Mechatronics Centre

- The history of equation-based modeling begins way before the invention of the first programming language.
- Although the term *object orientation* is a recent invention of computer science, its major concept can be traced back through centuries.
- The idea to compose a formal description of a system from its underlying objects is much older than computer science.
- So today is going to be a strange lecture in physics. We take a fresh look at the formulation of physical laws.

D'Alembert's Principle

Robotics and Mechatronics Centre

- It is a prerequisite for any object-oriented modeling approach that the behavior of the total system can be derived from the behavior of its components.
- A first manifestation of this problem can be found in the description of mechanical systems with rigidly connected bodies.

"Given is a system of multiple bodies that are arbitrarily [rigidly] connected with each other. We suppose that each body exhibits a natural motion that it cannot follow due to the rigid connections with the other bodies. We search the motion that is imposed to all bodies."

Jean-Baptiste le Rond d'Alembert, 1758

D’Alembert’s Principle

Robotics and Mechatronics Centre

- The method that leads to the solution of the problem is known today as d'Alembert's principle.
- His contribution is based, upon others, on the work of **Jakob I. and Daniel Bernoulli** and **Leonhard Euler**.
- It was brought to its final form by **Joseph-Louis de Lagrange** and is often presented today by the following equation:

$$\Sigma \mathbf{f} - m\mathbf{a} = 0$$

source: Wikimedia commons

Jean-Baptiste le Rond d’Alembert

D'Alembert's Principle

- The method that leads to the solution of the problem is known today as d'Alembert's principle.
- His contribution is based, upon others, on the work of **Jakob I. and Daniel Bernoulli** and **Leonhard Euler**.
- It was brought to its final form by **Joseph-Louis de Lagrange** and is often presented today by the following equation:

$$\Sigma \mathbf{f} - m\mathbf{a} = 0$$

source: Wikimedia commons

Jakob I. Bernoulli

D'Alembert's Principle

- The method that leads to the solution of the problem is known today as d'Alembert's principle.

- His contribution is based, upon others, on the work of **Jakob I. and Daniel Bernoulli** and **Leonhard Euler**.

- It was brought to its final form by **Joseph-Louis de Lagrange** and is often presented today by the following equation:

$$\Sigma \mathbf{f} - m\mathbf{a} = 0$$

source: Wikimedia commons

Daniel Bernoulli

D’Alembert’s Principle

- The method that leads to the solution of the problem is known today as d'Alembert's principle.
- His contribution is based, upon others, on the work of **Jakob I. and Daniel Bernoulli** and **Leonhard Euler**.
- It was brought to its final form by **Joseph-Louis de Lagrange** and is often presented today by the following equation:

$$\Sigma \mathbf{f} - m\mathbf{a} = 0$$

source: Wikimedia commons

Leonhard Euler

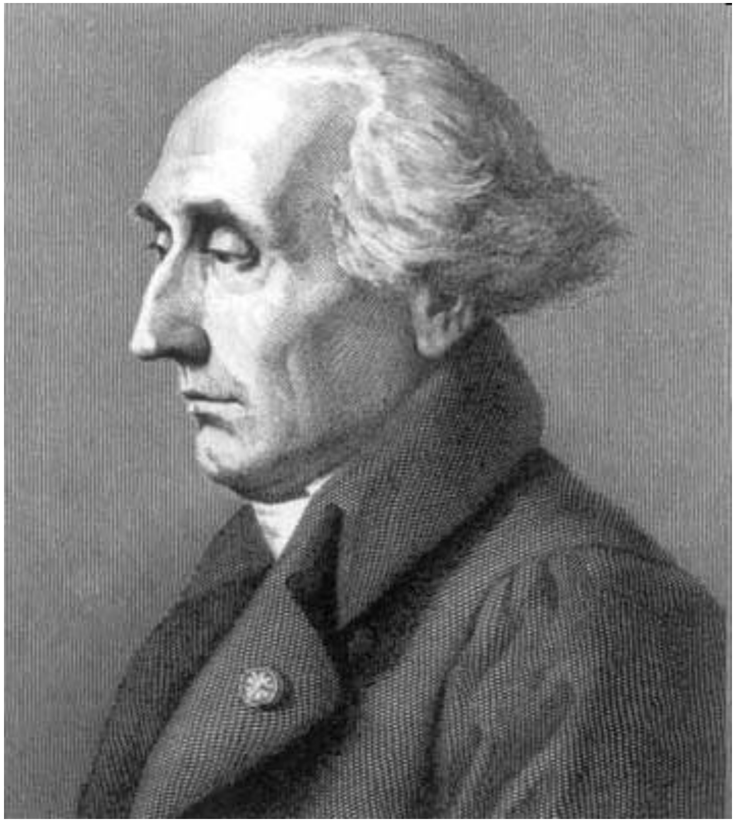
D’Alembert’s Principle

- The method that leads to the solution of the problem is known today as d'Alembert's principle.

- His contribution is based, upon others, on the work of **Jakob I. and Daniel Bernoulli** and **Leonhard Euler**.

- It was brought to its final form by **Joseph-Louis de Lagrange** and is often presented today by the following equation:

$$\Sigma \mathbf{f} - m\mathbf{a} = 0$$

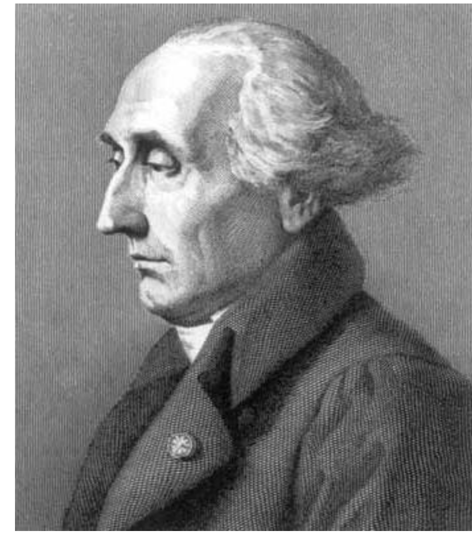


source: Wikimedia commons

Joseph-Louis de Lagrange

D’Alembert’s Principle



source: Wikimedia commons

1691 **1811**

- It took 120 years and the brainpower of the greatest mathematicians to bring d’Alembert’s Principle into its final form!

- 120 years for this equation: $\Sigma\mathbf{f} - m\mathbf{a} = 0$???

D’Alembert: The classic way

- Unjustifiably, the presentation of

$$\Sigma \mathbf{f} - m\mathbf{a} = 0$$

reduces a major mechanical principle to a trivial equation.

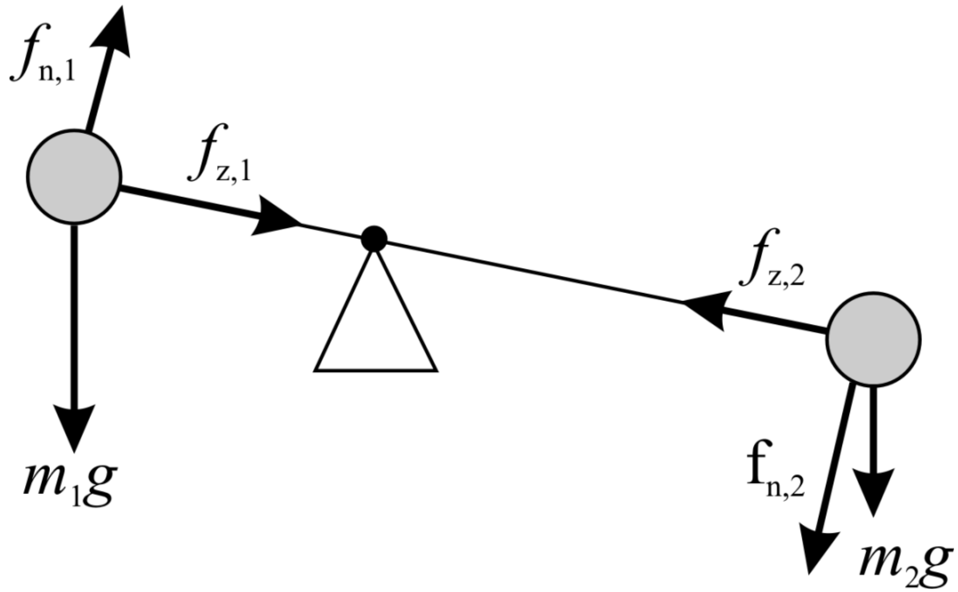
- Often it is mistakenly “derived” by transforming Newton´s law $\mathbf{f} = m\mathbf{a}$, but Newton’s law holds just for a single point of mass.

- D´Alembert’s principle applies to complete mechanic systems. Its central idea is to take the imposed motion as counteracting force.

- D’Alembert’s principle is best understood by applying it to an example…

D'Alembert: The classic way

- Let us model this asymmetric seesaw. l_1 and l_2 denote the lengths of the opposing lever arms.



- We start by the equations for the lever arm.

- Relation of velocity (in direction of $\mathbf{e}_n$, normal to the lever arm):

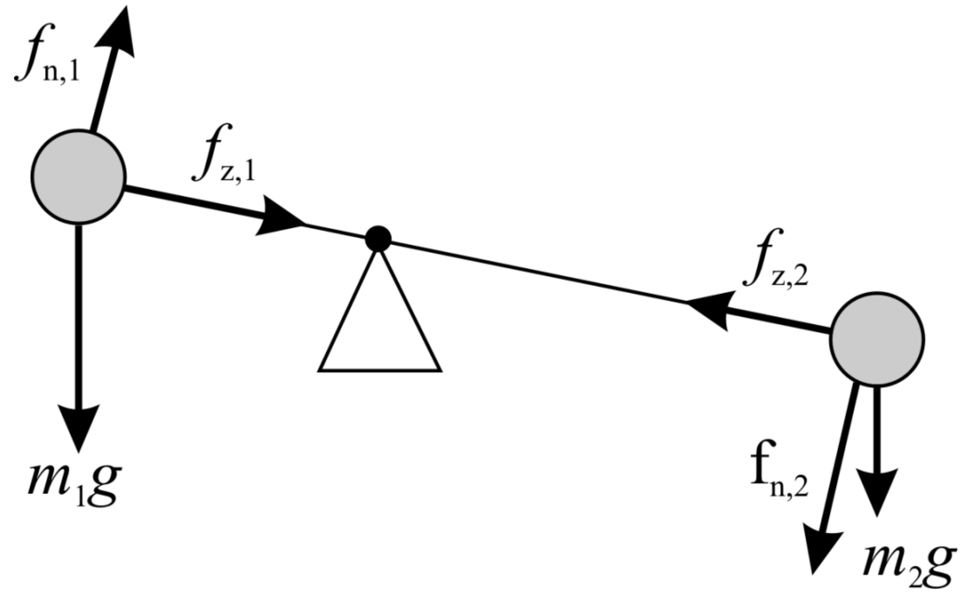
$$v_1 \cdot l_2 = -v_2 \cdot l_1$$

- Balance of force:

$$f_{n,1} \cdot l_1 + f_{n,2} \cdot l_2 = 0$$

D'Alembert: The classic way

- Each body element defines one differential equation since the acceleration is the time-derivative of the velocity.



- Left Body:

$$dv_1/dt = a_1$$

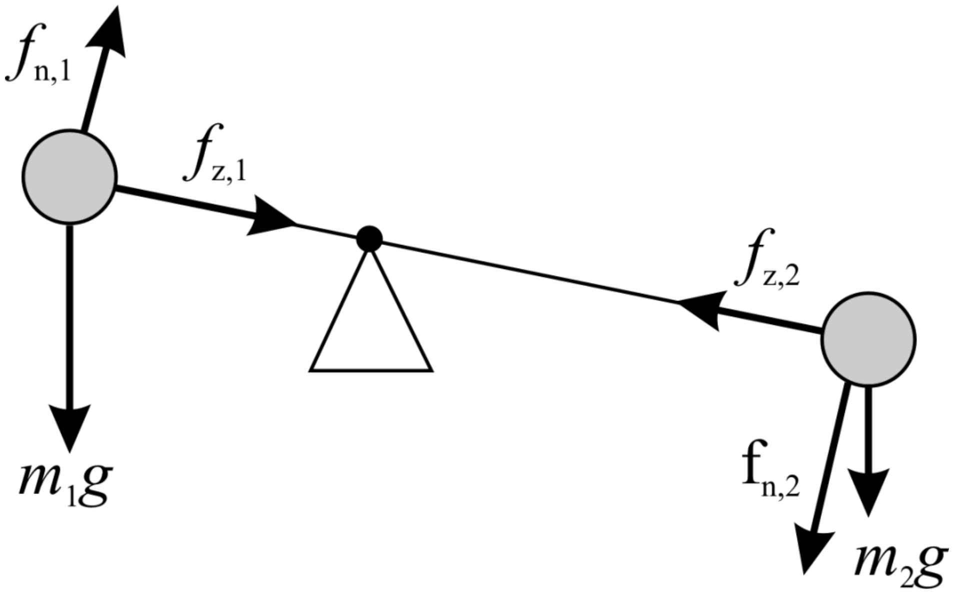
- Right Body:

$$dv_2/dt = a_2$$

D'Alembert: The classic way

- D'Alembert's Principle can now be directly applied on the body components.

- The sum of all forces has to be in equilibrium with the imposed motion



- Left Body:

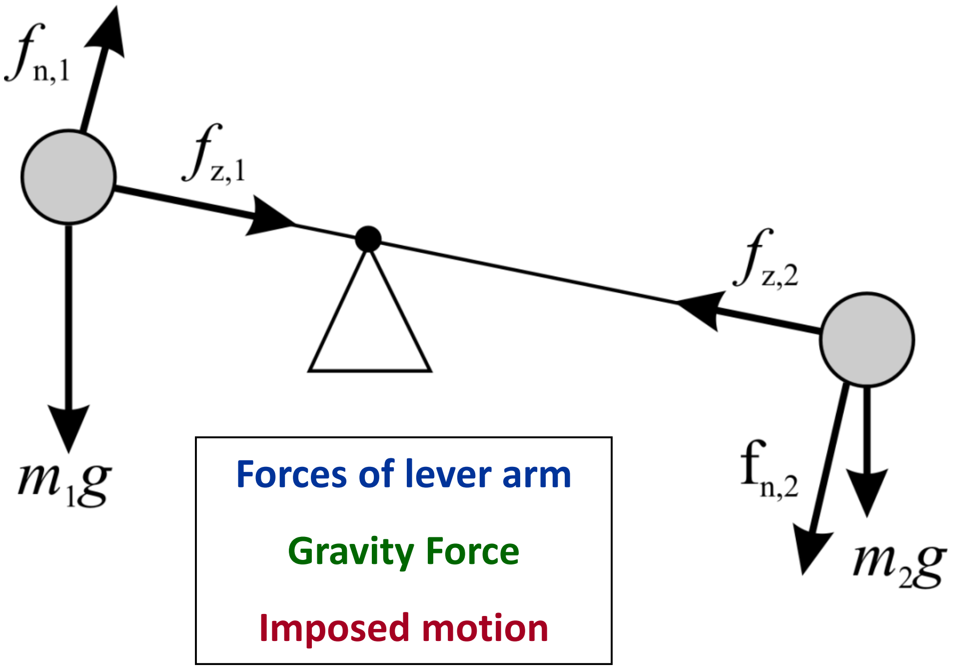
$$f_{n,1}\mathbf{e}_n + f_{z,1}\mathbf{e}_z + (0, -m_1g)^T - m_1a_1\mathbf{e}_n = \mathbf{0}$$

- Right Body:

$$f_{n,2}\mathbf{e}_n + f_{z,2}\mathbf{e}_z + (0, -m_2g)^T - m_2a_2\mathbf{e}_n = \mathbf{0}$$

D'Alembert: The classic way

- D'Alembert's Principle can now be directly applied on the body components.



- The sum of all forces has to be in equilibrium with the imposed motion

- Left Body:

$$f_{n,1}\mathbf{e}_n + f_{z,1}\mathbf{e}_z + (0, -m_1 g)^T - m_1 a_1 \mathbf{e}_n = \mathbf{0}$$

- Right Body:

$$f_{n,2}\mathbf{e}_n + f_{z,2}\mathbf{e}_z + (0, -m_2 g)^T - m_2 a_2 \mathbf{e}_n = \mathbf{0}$$

D'Alembert: The classic way

Robotics and Mechatronics Centre

- In total, we have 8 unknowns: a_1,a_2, v_1, v_2, $f_{n,1}$, $f_{n,2}$ $f_{z,1}$, $f_{z,2}$
- And 8 (4 + 2·2) scalar differential-algebraic equations:

$$v_1 \cdot l_2 = -v_2 \cdot l_1$$

$$f_{n,1} \cdot l_1 + f_{n,2} \cdot l_2 = 0$$

$$\mathrm{d}v_1/\mathrm{d}t = a_1$$

$$\mathrm{d}v_2/\mathrm{d}t = a_2$$

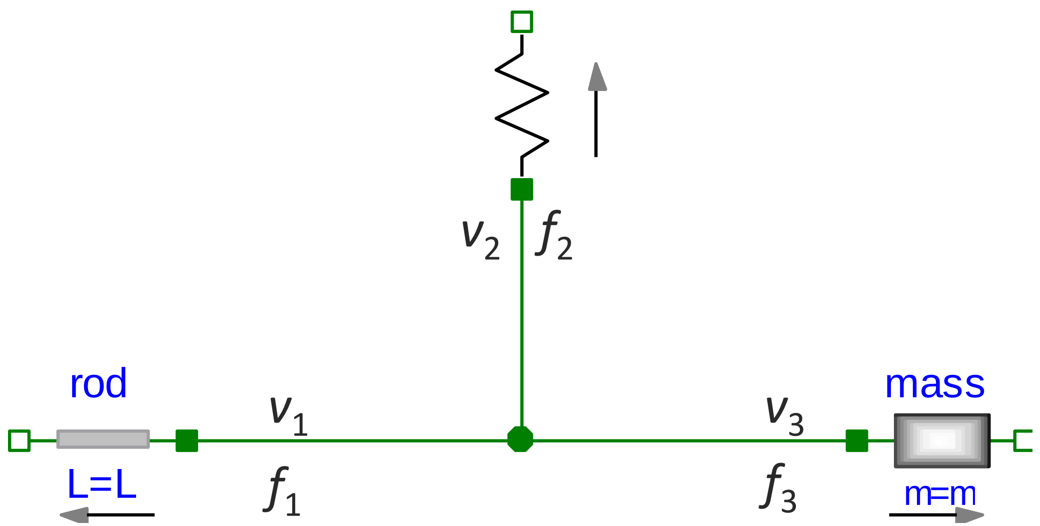
$$f_{n,1}\mathbf{e}_n + f_{z,1}\mathbf{e}_z + (0, -\mathrm{m}_1\mathrm{g})^T - m_1 a_1 \mathbf{e}_n = \mathbf{0} \text{ (2 scalar equations)}$$

$$f_{n,2}\mathbf{e}_n + f_{z,2}\mathbf{e}_z + (0, -\mathrm{m}_2\mathrm{g})^T - m_2 a_2 \mathbf{e}_n = \mathbf{0} \text{ (2 scalar equations)}$$

- So the system is complete and regular. Mission accomplished.

D'Alembert: The node equations

There is a different perspective on D´Alembert´s Principle

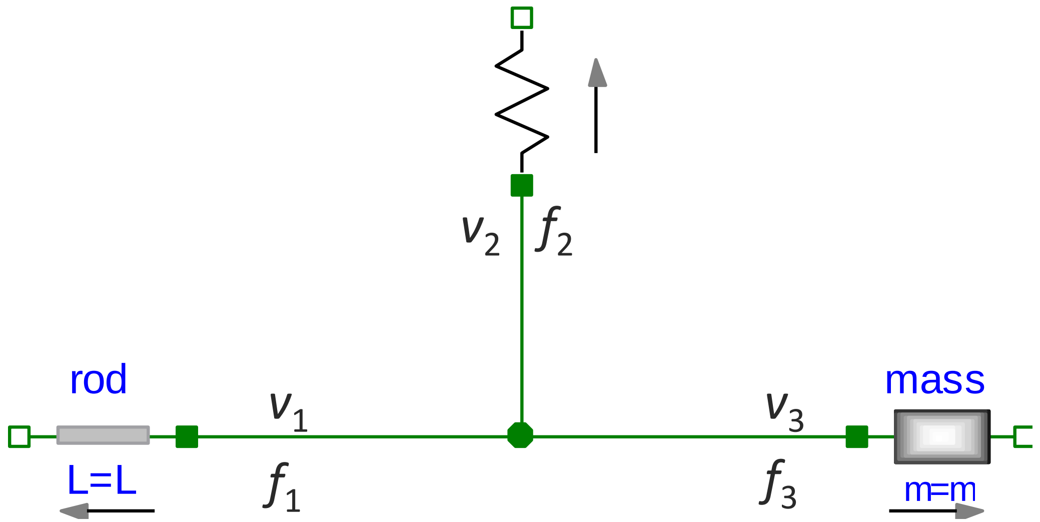


- Let us look at a mechanical node (or flange, if you prefer) that rigidly connects different mechanical components.
- Each component defines its own velocity v_1, v_2, …, v_n and its own force f_1, f_2, …, f_n.

D'Alembert: The node equations

Robotics and Mechatronics Centre

No we can state the following equations for this node:



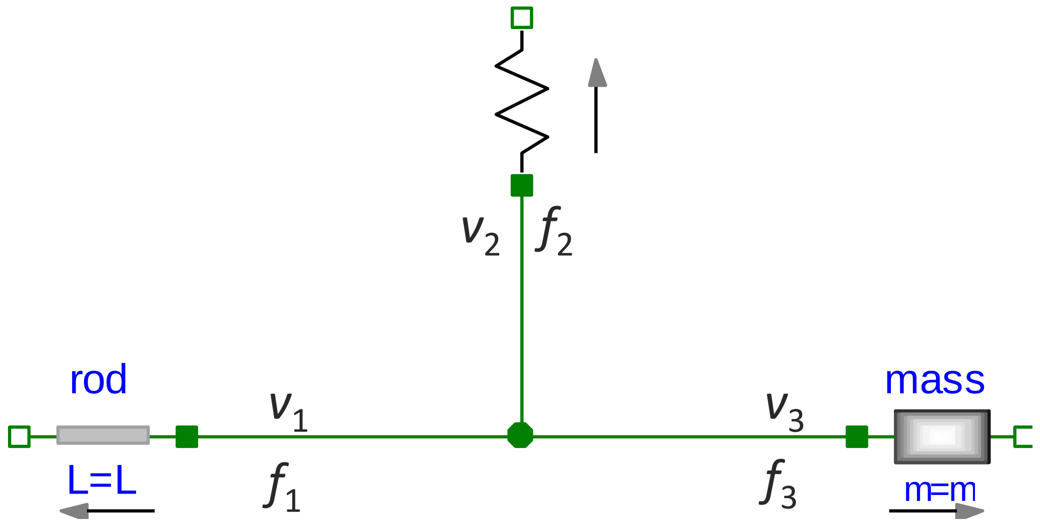
- Since the connection is rigid, all velocities must be equal:

$$v_1 = v_2 = \ldots = v_n$$

- And d'Alembert's principle is telling us that there is a balance of force:

$$f_1 + f_2 + \ldots + f_n = 0$$

D'Alembert: The node equations



- If we do so, the body equations are represented by:

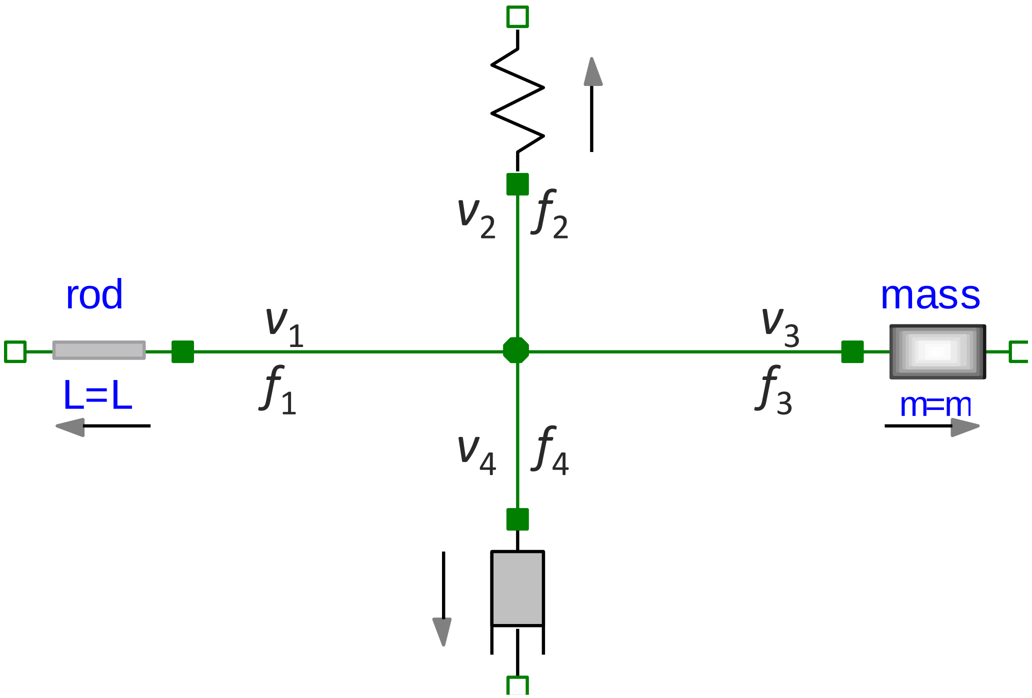
$$dv/dt = a$$

$$f = ma$$

D'Alembert: The node equations

Robotics and Mechatronics Centre

When we add another component to the node....



- ...only the equations of the node change, but the equations of the individual components remain untouched.

D`Alembert`s Principle: Summary

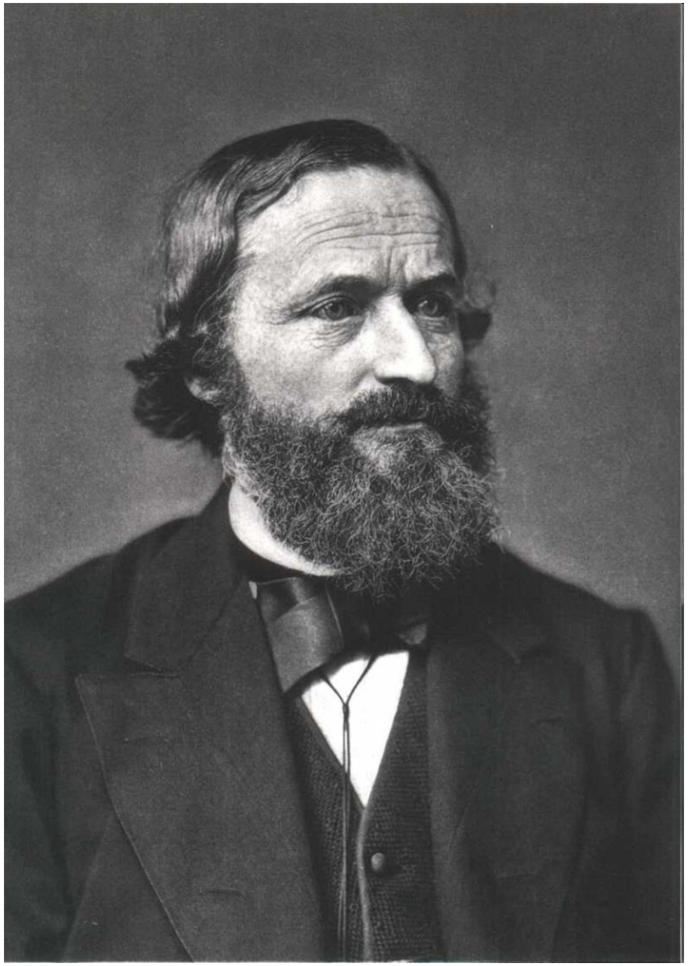
- D'Alembert's principle is not an actual physical law. It represents a methodology to obtain a correct set of differential-algebraic equations for arbitrary mechanical systems.

- D'Alembert's principle reveals itself to be simple and elegant for this purpose, but it is by no means a triviality.

Kirchhoff`s Circuit Laws

- Whereas D'Alembert's principle provides a method to derive a correct set of equations for rigidly constrained mechanical components, *Gustav Kirchhoff* accomplished a similar task for the electrical domain.

- In 1845, he stated his two famous circuit laws.



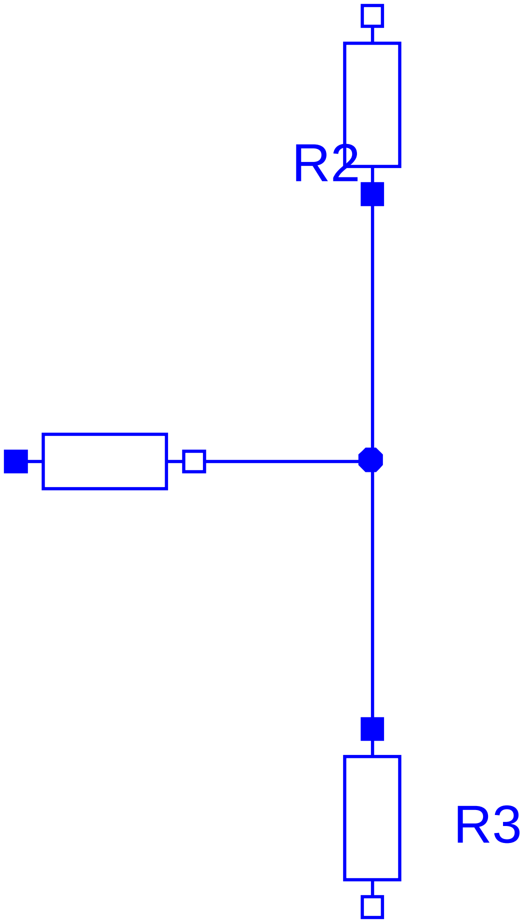
Gustav Robert Kirchhoff
1824 - 1887

The 1st Circuit Law

- The first circuit law states that for each electrical node, the sum of the incoming currents must equal the sum of the outgoing currents.

$$\Sigma\, i_{in} = \Sigma\, i_{out}$$

- Unfortunately, it not always clear in what direction the current is flowing.

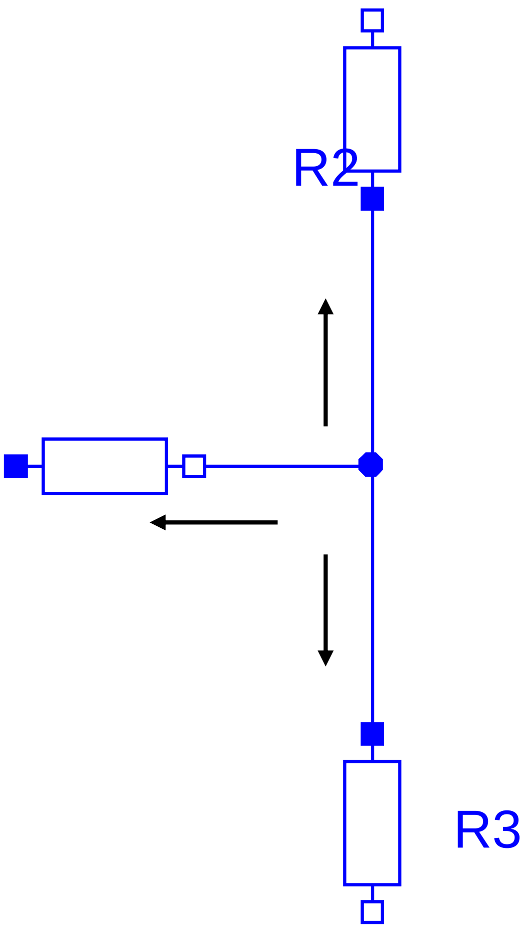


The 1st Circuit Law

- Fortunately, we can transform this law into a more convenient form, by defining the flow direction and allowing negative currents.

- If we define that the current always flows from the node into the components, we can state:

$$\Sigma\, i_n = 0$$



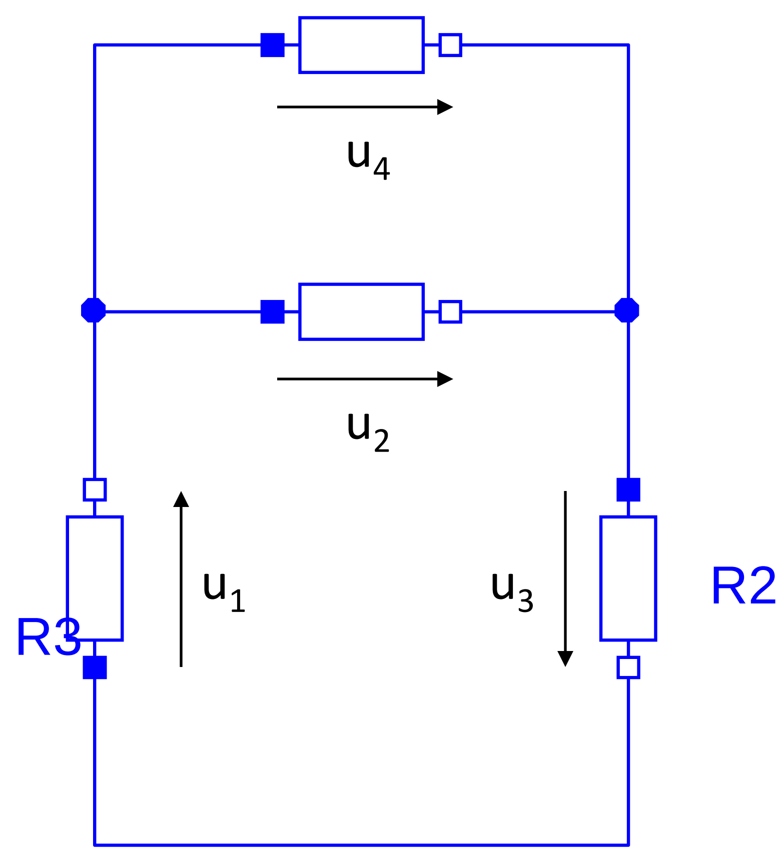
The 2nd Circuit Law

Robotics and Mechatronics Centre

- The second circuit law is the mesh (or loop) rule.
- It states that the directed sum of the electrical voltages around any closed circuit must be zero.

$$\Sigma u_n = 0$$

- This form is rather inconvenient since it requires to decompose the electric circuit into its loops.

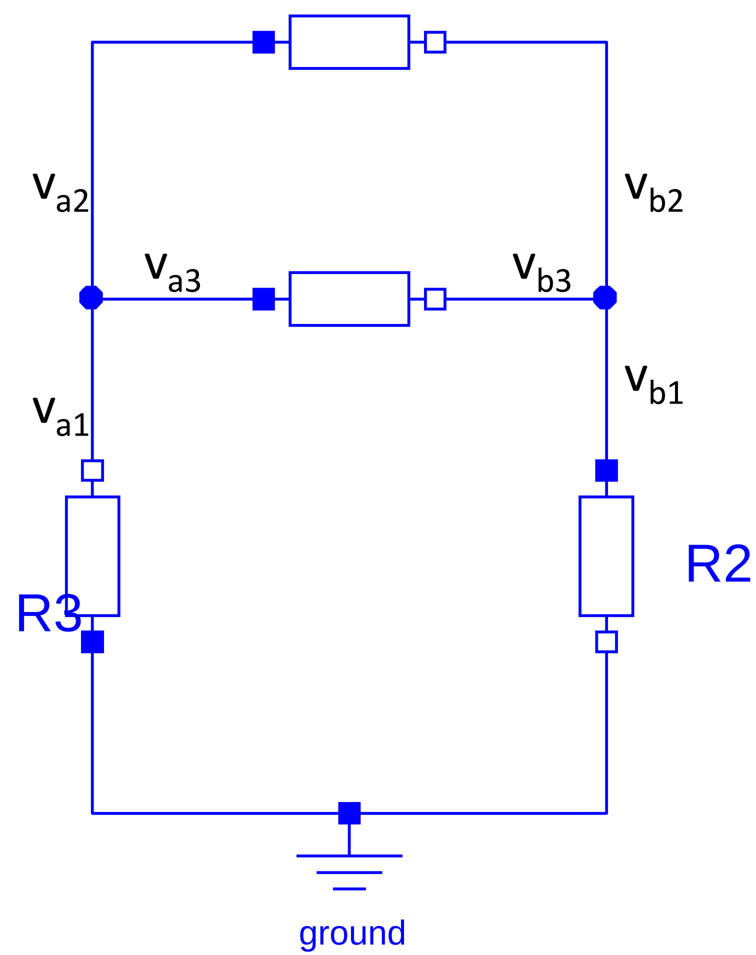


The 2nd Circuit Law

- Also this rule can be transformed into a more convenient form.
- To this end, we ground the circuit.
- Now, we can assign an electric potential v (Spannungspotential) to each node .
- Kirchhoff's mesh rule is now equivalent to the node equation

$$v_1 = v_2 = \ldots = v_n$$

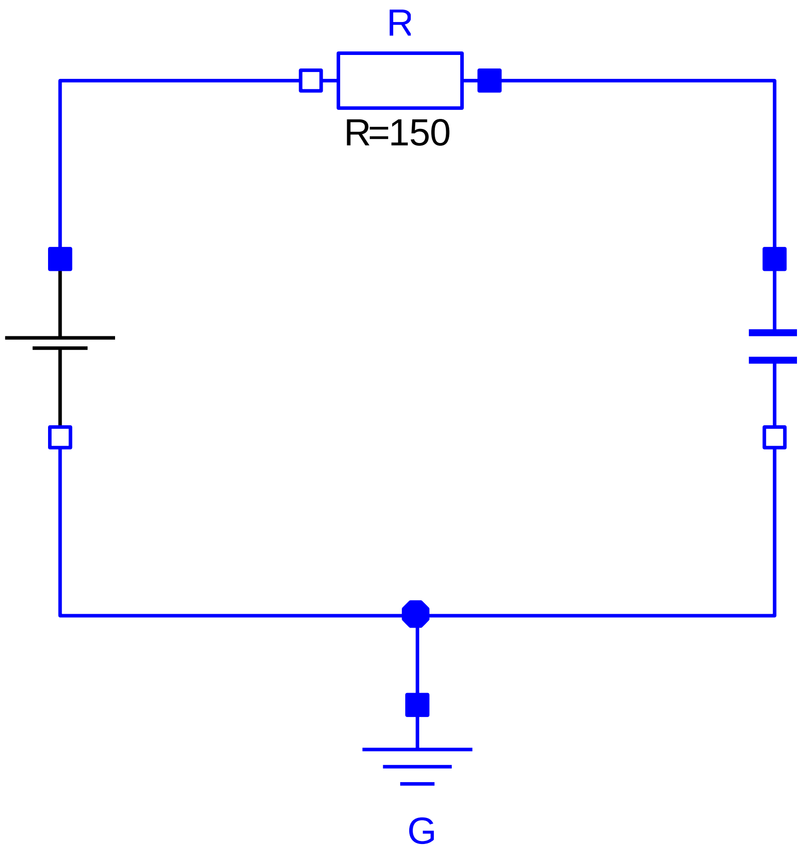
- The voltage potentials at each node must be equal.



Kirchhoff's Laws in Action

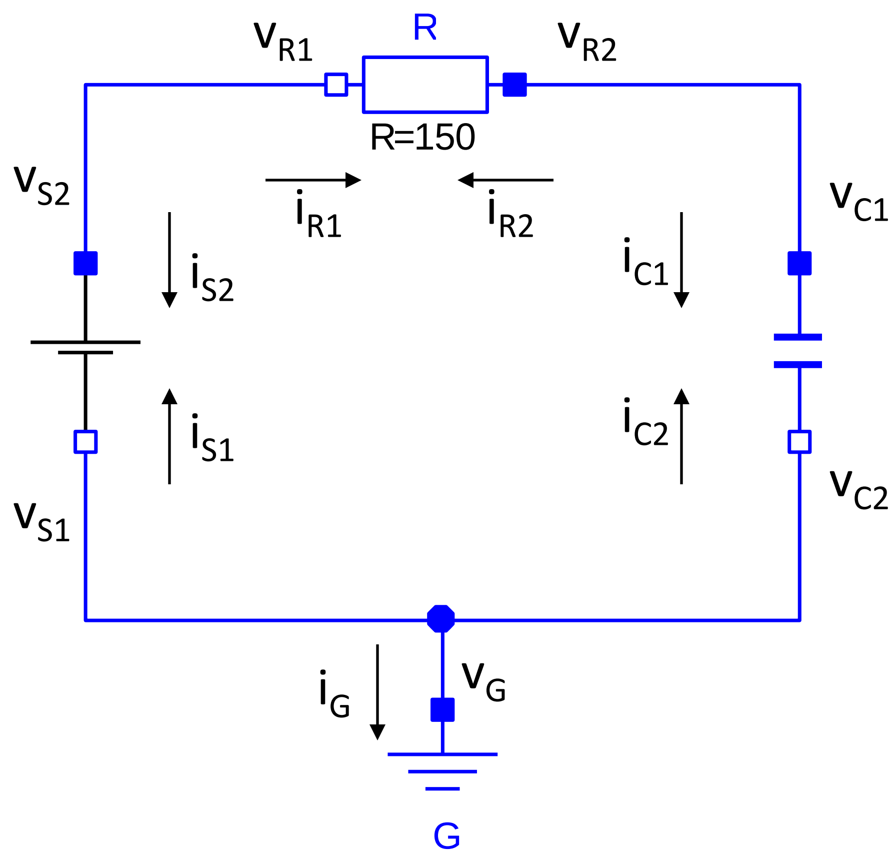
Robotics and Mechatronics Centre

Let us model a simple electric circuit:



Kirchhoff's Laws in Action

Let us model a simple electric circuit:

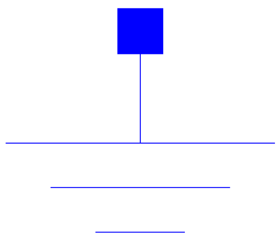


Kirchhoff's Laws in Action

First we start with the component equations

- The grounding is easy
  (2 unknowns, 1 equation):

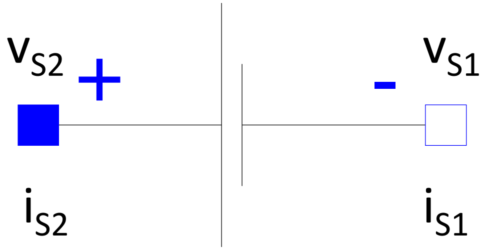
$$V_G = 0$$



- The voltage source connects two nodes:
  (4 unknowns, 2 equations)

$$i_{S1} + i_{S2} = 0$$

$$v_{S1} + 10V = v_{S2}$$



Kirchhoff`s Laws in Action

Robotics and Mechatronics Centre

First we start with the component equations

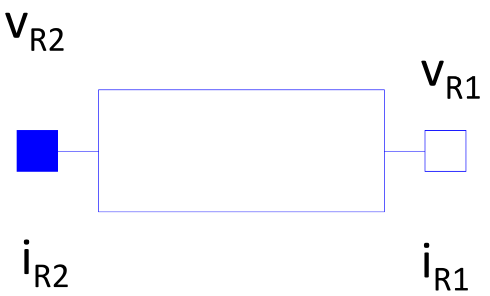
- The resistor is modeled by famous Ohm´s law: (5 unknowns, 3 equations)

$$u_R = R*i_{R1}$$

with

$$i_{R1} + i_{R2} = 0$$

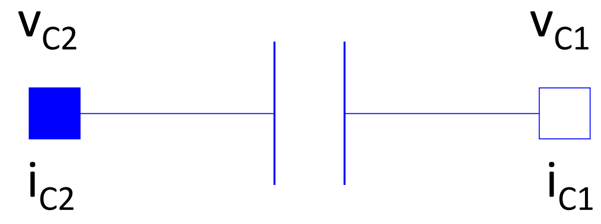
$$v_{R1} + u_R = v_{R2}$$



Kirchhoff`s Laws in Action

First we start with the component equations

- The capacitor contains a differential equation. The voltage is induced by a charge. The derivative of the charge is the current. (5 unknowns, 3 equations)



$$C * du_C/dt = i_{C1}$$

with

$$i_{C1} + i_{C2} = 0$$

$$v_{C1} + u_C = v_{C2}$$

Kirchhoff`s Laws in Action

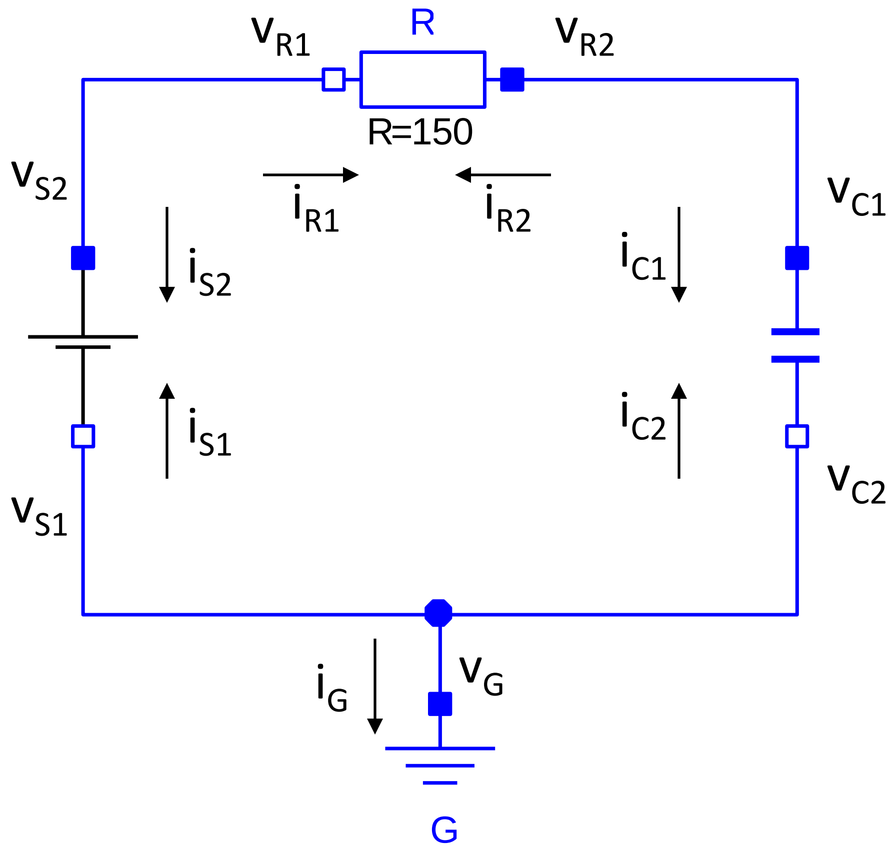
Robotics and Mechatronics Centre

Then we continue by applying
Kirchhoff's law for each node:

$$v_{S2} = v_{R1}$$
$$i_{S2} + i_{R1} = 0$$

$$v_{R2} = v_{C1}$$
$$i_{R2} + i_{C1} = 0$$

$$v_{C2} = v_G$$
$$v_{S1} = v_G$$
$$i_{C2} + i_{S1} + i_G = 0$$



Kirchhoff`s Laws in Action

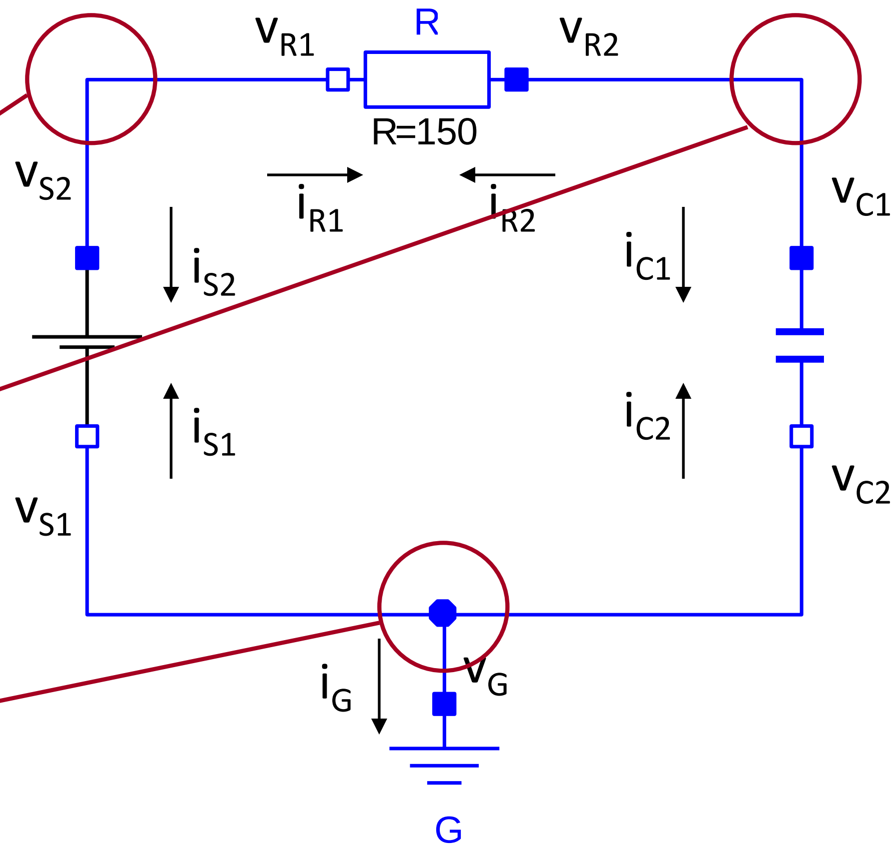
Robotics and Mechatronics Centre

Then we continue by applying
Kirchhoff's law for each node:

$$v_{S2} = v_{R1}$$
$$i_{S2} + i_{R1} = 0$$

$$v_{R2} = v_{C1}$$
$$i_{R2} + i_{C1} = 0$$

$$v_{C2} = v_{G}$$
$$v_{S1} = v_{G}$$
$$i_{C2} + i_{S1} + i_{G} = 0$$



Kirchhoff`s Laws in Action

Robotics and Mechatronics Centre

When we collect all equations, we count 16 equations and 16 unknowns.
The system of differential-algebraic equations is complete.

$$v_{S2} = v_{R1}$$
$$i_{S2} + i_{S1} = 0$$

$$v_{R2} = v_{C1}$$
$$i_{R2} + i_{C1} = 0$$

$$v_{C2} = v_G$$
$$v_{S1} = v_G$$
$$i_{C2} + i_{S1} + i_G = 0$$

Node equations

$$v_G = 0$$

$$i_{S1} + i_{S2} = 0$$
$$v_{S1} + 10V = v_{S2}$$

$$u_R = R*i_{R1}$$
$$I_{R1} + I_{R2} = 0$$
$$v_{R1} + u_R = v_{R2}$$

$$C*du_C/dt = i_{C1}$$
$$I_{C1} + I_{C2} = 0$$
$$v_{C1} + {}_{uC} = v_{C2}$$

Component Equations

Object-Orientation

In this way, Kirchhoff enabled the object-oriented modeling of electric systems.

- By having general laws for the junctions between components, the equations of the individual components become ***generally applicable and reusable***.
- Kirchhoff's laws prove that the junction structure of an electrical circuit provides a general interface for all potential electric components. The implementation of a component (its internal equations) can therefore be ***separated from the interface*** (its nodes).
- The interface of a component describes how the components can be applied, whereas the implementation describes what is its internal functionality. Components with equivalent interface can be ***generically interchanged***.
- Known circuits can be ***extended*** by adding further junctions and components. Knowledge can be ***inherited***.

Object-Orientation

- The highlighted terms represent keywords or motivations common to object-oriented programming.
- Next week, we are going to see how the modeling perspective of object-orientation is realized within a computer language.

Questions?