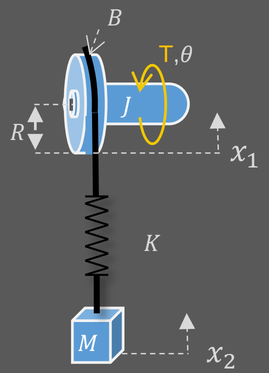
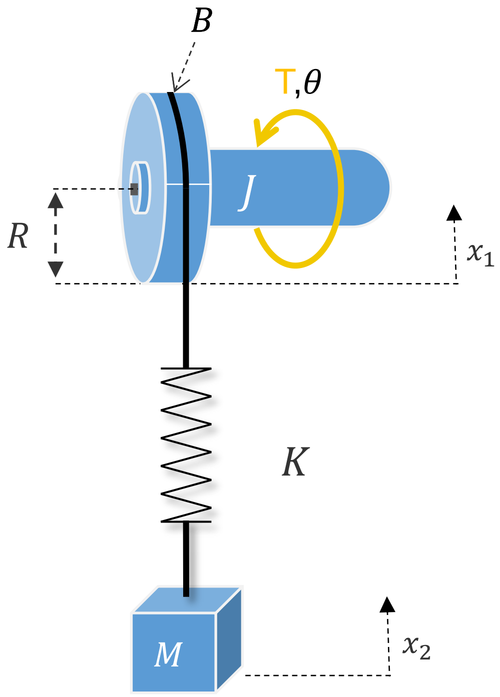
Modeling and Simulation of a rotational-translational system using solidThinking Activate



Rotational-translational system

Converting rotational motion to translational motion with a implemented damping factor due to friction.



Objectives

- Learn how to…
  - … convert translational motion to rotational motion (and counter wise)
  - … create block diagram with “MathExpressions”-Block
  - … validate the results with help of transfer functions
    - … by separating the model into several feedback-loops and thus simplifies the block diagram

Creation and Simulation of a rotational-translational system

Theoretical background and how to implement it using *Activate*

Step 1: Construction of equations

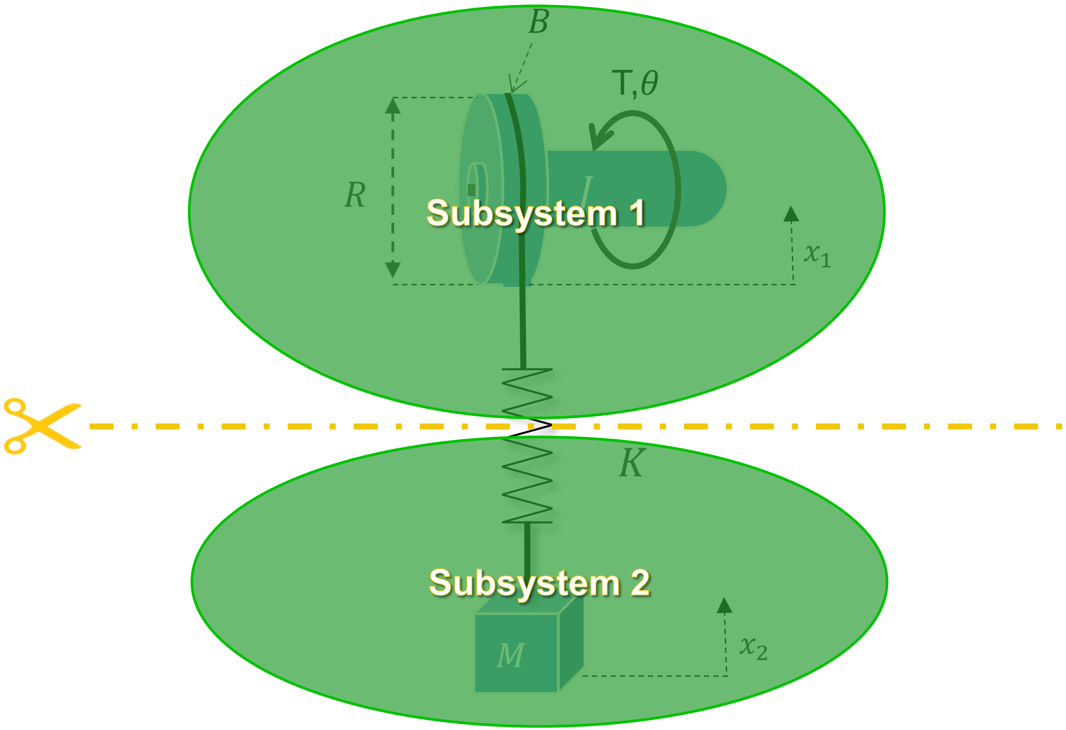
- Step 1.1: Creating of subsystem
- Step 1.2: Introduction of cut-forces

Step 2: Implementation using *Activate*

Step 3: Validation of the results

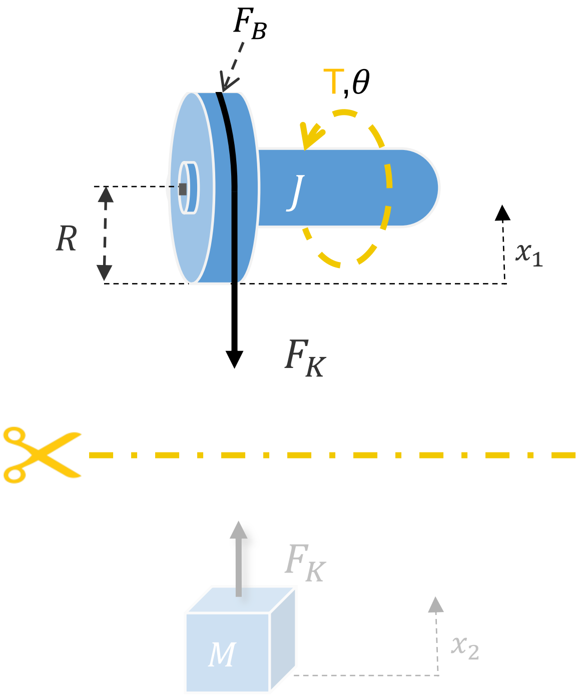
Step 1: Construction of equations

- Step 1.1: Creating subsystems



Step 1: Construction of equations *(Subsystem 1)*

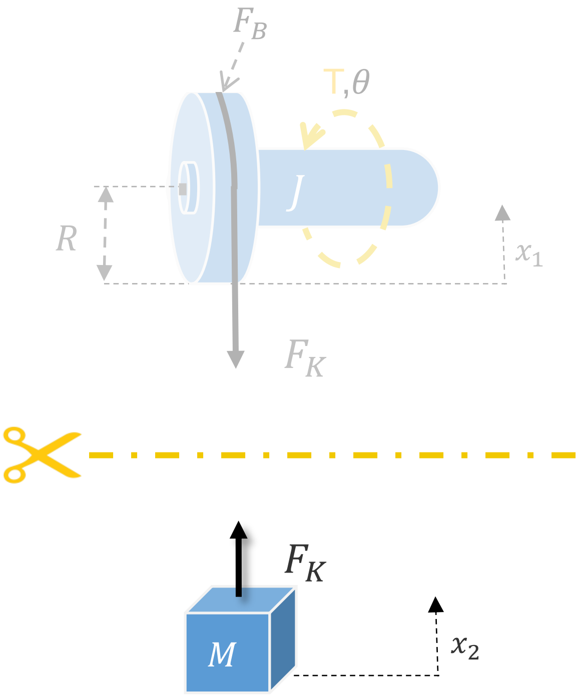
- Step 1.2: Introduction of cut-forces
- Equations:
  - $Friction = Damping\ (simplified): F_B = B\dot{x}_1;$
  - $T = FR$
  - Coupling Term: $x_1 = \theta R$
  - $J\ddot{\theta} = \mathrm{T} - K(x_1 - x_2)\ - \mathrm{B}\dot{x}_1$



Step 1: Construction of equations *(Subsystem 1)*

- Step 1.2: Introduction of cut-forces
- Equations:

  - Coupling Term: $x_1 = \theta R$
  - $M\ddot{x}_2 = K(x_1 - x_2)$



Step 2: Implementation using *Activate*

- External input signal is a unit torque impulse.
  - Applying a step torque input to this system will produce a displacement value that goes to infinity (Fig.1)
  - We will apply the following unit impulse torque (Fig. 2) which will produce a finite displacement value (Fig.3).

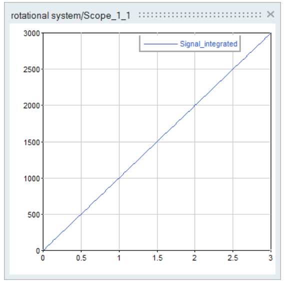


Fig. 1: Integrated step signal

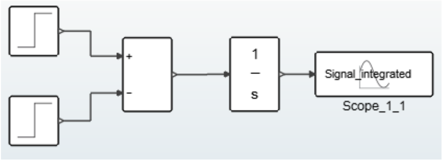


Fig. 2: Activate model of unit impulse torque and integration

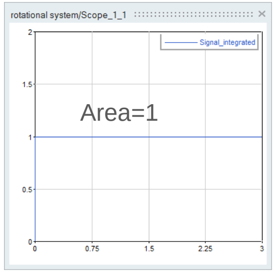
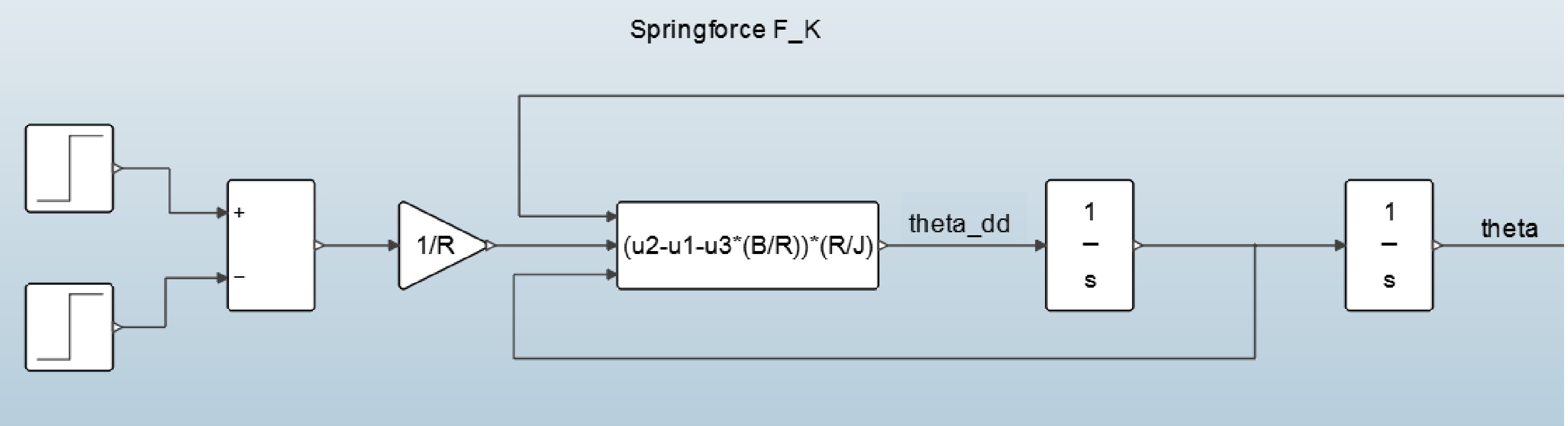


Fig. 3: Integrated unit impulse signal

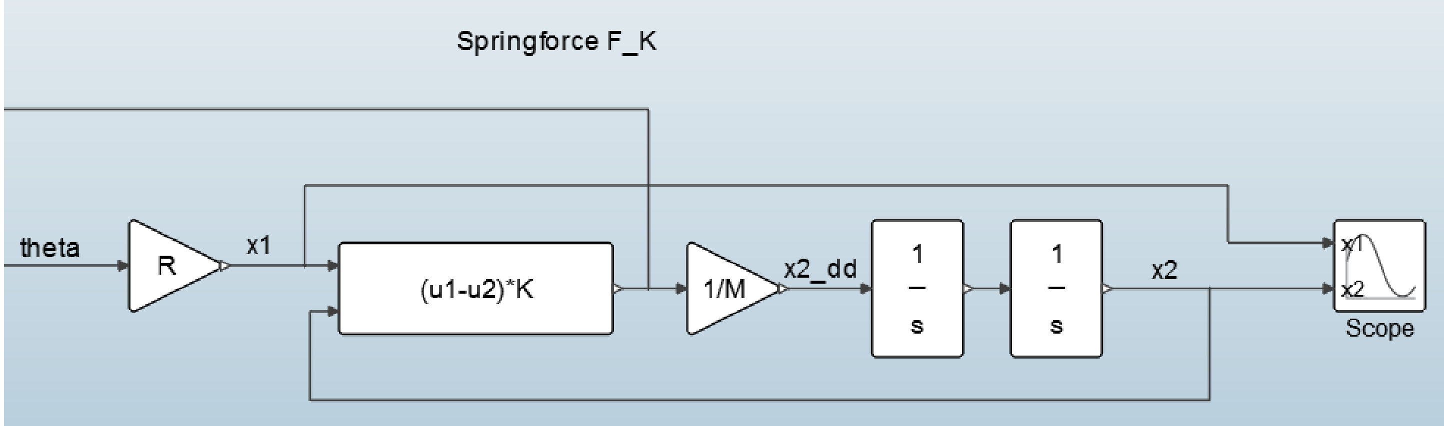
Step 2: Implementation using *Activate*

- *Subsystem 1:*
- External input signal is a unit torque impulse.
  - Divide torque by radius R to obtain force input
  - Convert force to angular acceleration using $\frac{R}{J}$
  - Use "MathExpressions"-Block to directly put in the equations of motion: $J\ddot{\theta} = \mathrm{T} - K(x_1 - x_2) - \mathrm{B}\dot{x}_1$



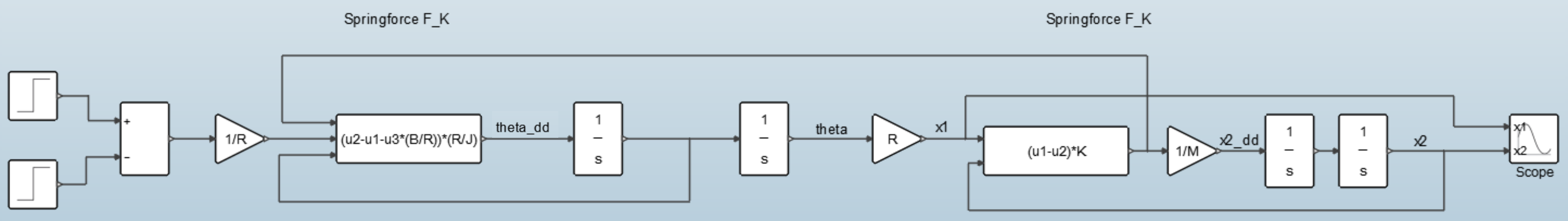
Step 2: Implementation using *Activate*

- *Subsystem 2:*
- Translate motion from translational to rotational (or counter wise) using a coupling term to convert the signal: $x_1 = \theta R$.



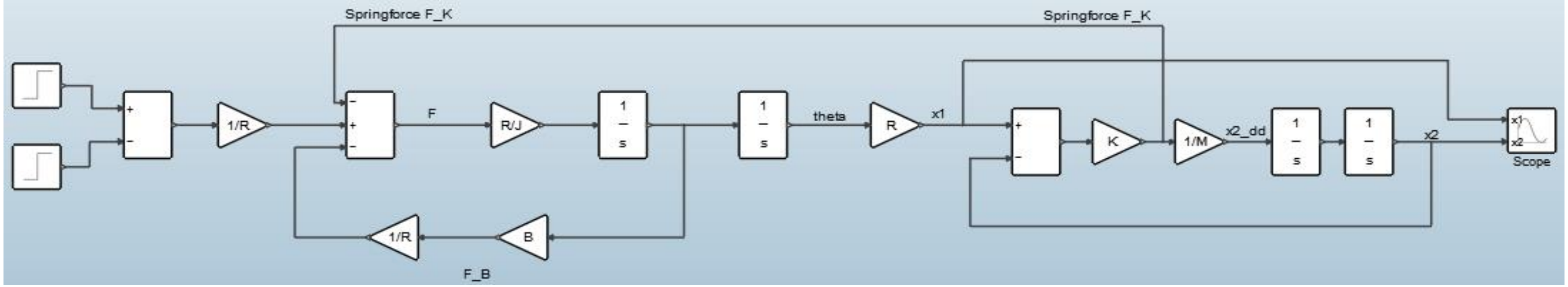
Step 2: Implementation using *Activate*

- Completed model with “MathExpressions”-Block



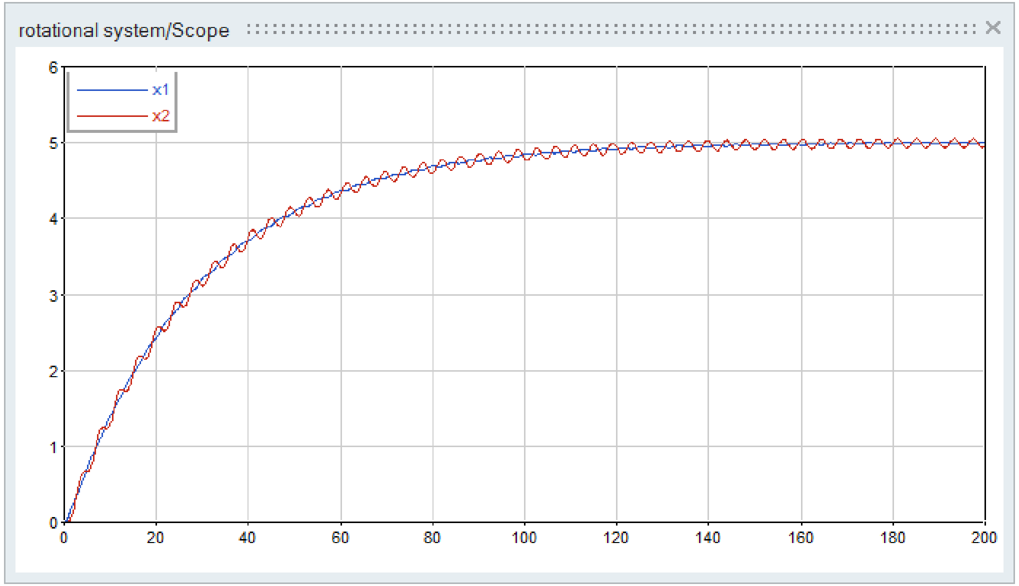
=

- Common way of the model with gained feedback loops



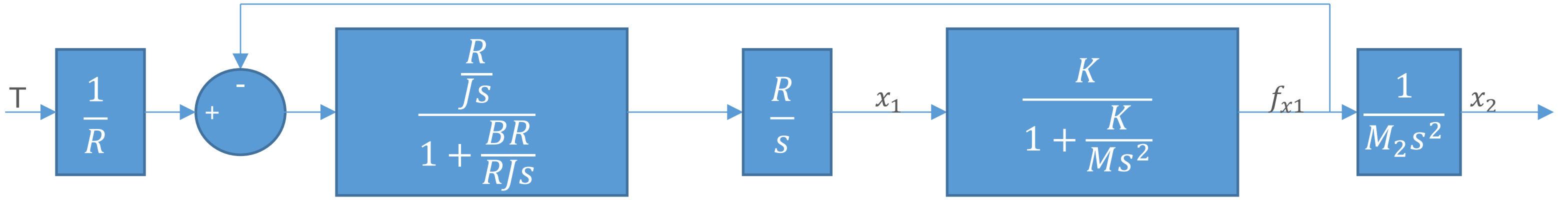
Step 3: Validation of the results

- The rotational system oscillates in time response at $x_1 = x_2 = 5$
- $\frac{R}{B} = 5$; → Validation via transfer functions



Step 3: Validation of the results

- Validation via transfer functions (TF)
  - Simplify Block Diagram



- TF of entire system

$$\frac{x_1}{T} = \frac{1}{R}\,\frac{\frac{R^2}{s(Js+B)}}{1+\frac{R^2 K M s^2}{s(Js+B)(Ms^2+K)}}$$

- *in steady state (s→0):*

$$\frac{x_1}{T} \cong \frac{RK}{sBK} \cong \frac{R}{sB}$$

- *Applying a unit impulse torque input:*

$$x_1 \cong \frac{0.4}{0.08} \cong 5 \;☑$$