# Chapter 3

# Modeling for control: 2R robot

Here we take a look at a straightforward example of modeling for control, and we'll consider a planar collaborative robot arm with two revolute joints. The aim is to show why do we need analytical modeling to calculate control signals and how mechanical structure affects control algorithms. The section is indented to give the general idea of modeling for control of robots for physical Human-Robot-Environment Interaction (pHREI) as systems.

## 3.1 Mechanical structure and kinematics

The schematic representation of the 2R planar robot[1] is shown in Fig. 3.1. It consists of two rigid links with lengths $l_1$ and $l_2$, point masses $m_1$ and $m_2$, placed in the middle of corresponding link, and two revolute joints $A$ and $B$ without any elastic elements. Joint $A$ connects the robot to the ground. $\Psi_0$ represents a fixed base that is coincided with the first link's frame $\Psi_1$ in an initial position. $\Psi_2$ is the frame of the second link $BC$, $\Psi_3$ is the frame of point $C$ that we treat as end-effector in this example. Since it is a planar mechanism, the end-effector $C$ is able to travel in the $xy$ plane, revolute joints rotate around the $z$ axis, and gravity $g$ is acting in the $-y$ direction.

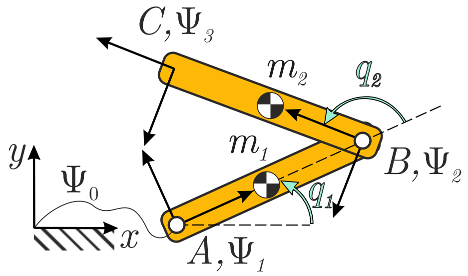



Figure 3.1: Schematic representation of the 2R planar robot

To describe points on a plane, we need to choose a point in the space as an origin of a coordinate system, two orthogonal coordinate axes $x$ and $y$, and use coordinates frames to associate coordinates/numbers to points. In the example, it is convenient to take point $A$ as an origin point and frame $\Psi_0$ as a frame fixed in space.

The *forward kinematics* refers to the map associating of the given joints coordinates $q$ to the position and orientation of the robot's end-effector [5][2]. The Cartesian coordinates $(C_x, C_y)$ of point $C$ expressed in frame $\Psi_0$ can be calculated as functions of joint angles $(q_1, q_2)$:

$$C_x = l_1 \cos q_1 + l_2 \cos(q_1 + q_2),$$
$$C_y = l_1 \sin q_1 + l_2 \sin(q_1 + q_2),$$

the current position $p$ of the end-effector $C$ in matrix form is

$$p := \begin{pmatrix} C_x \\ C_y \end{pmatrix} = \begin{pmatrix} l_1 \cos q_1 + l_2 \cos(q_1 + q_2) \\ l_1 \sin q_1 + l_2 \sin(q_1 + q_2) \end{pmatrix}. \tag{3.1}$$

[1]2R planar robot is the one with two revolute joints

[2]Here we treat point $C$ with frame $\Psi_3$ that interacts with the environment as an end-effector, but generally, an end-effector is a device or tool that's connected to the end of a robot arm

In this planar example we use an *explicit parametrization of the space*, since we use $n$ coordinates of a single point $C$, that we treat as the end-effector, for a $n$-dimensional space. In this case the velocity $v$ of point $C$[3] is just the time derivate of the corresponding coordinates (eq. 3.1)

$$v = \dot{p} = \begin{pmatrix} -l_1 \sin q_1 - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos q_1 + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix} = J(q)\dot{q}, \tag{3.2}$$

where $\dot{q}$ is a vector of links' angular velocities and $J(q)$ is the analytical Jacobian of (3.1)

$$J(q) = \begin{pmatrix} -l_1 \sin q_1 - l_2 \sin(q_1 + q_2) & -l_2 \sin(q_1 + q_2) \\ l_1 \cos q_1 + l_2 \cos(q_1 + q_2) & l_2 \cos(q_1 + q_2) \end{pmatrix}. \tag{3.3}$$

## 3.2 Dynamics

The *forward dynamics* is finding the robot's acceleration $\ddot{q}$ for a given the state $(q, \dot{q})$, the joints forces $F$ and torques $\tau$. The *inverse dynamics* is the problem of determining the joint forces $F$ and torques $\tau$ corresponding to the robot's state $(q, \dot{q})$ and acceleration $\ddot{q}$.

There are two common ways to analyze dynamics: the *Newton-Euler formulation* to describe the dynamics of rigid bodies and the *Lagrangian formulation* using the kinetic and potential energy of a robot.

### Lagrangian formulation

The configuration of the 2R planar robot can be described by the set of independent generalized coordinates $q \in \mathbb{R}^n$: $q_1$ for the first link $AB$ and $q_2$ for the second link $BC$ with respect to the first one. Each generalized coordinate $q_i$ is paired with the *generalized force or torque* $F_i, \tau_i \in \mathbb{R}^n$ applied to the corresponding link $i$.

The generalized coordinates $q$ represent points in *configuration space* [4]. Their rates $\dot{q}$ form vectors in *vector space* $\dot{q} \in \mathcal{V}$ that are attached at a position $q$. We introduce generalized torques $\tau$ (and generalized forces $F$ for linear motion) as linear operators which map a vector of velocities $\dot{q}$ to power $P \in \mathbb{R}$ and we call those co-vectors that are elements of the dual space:

$$\tau : \mathcal{V} \to \mathbb{R}.$$

No extra structure like an inner product is needed to look at power:

$$\tau(\dot{q}) \to \begin{pmatrix} \tau_1 & \tau_2 & \dots & \tau_n \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dots \\ \dot{q}_n \end{pmatrix}.$$

A Lagrangian function is the difference between the kinetic energy $\mathcal{K}(q, \dot{q})$ and the potential energy $\mathcal{P}(q)$

$$\mathcal{L}(q, \dot{q}) = \mathcal{K}^*(q, \dot{q}) - \mathcal{P}(q). \tag{3.4}$$

The total kinetic and potential energies

$$\mathcal{K}(q, \dot{q}) = \mathcal{K}_1(q_1, \dot{q}_1) + \mathcal{K}_2(q_2, \dot{q}_2) = \frac{m_1 v_1^2}{2} + \frac{m_2 v_2^2}{2}, \tag{3.5}$$

$$\mathcal{P}(q) = \mathcal{P}_1(q_1) + \mathcal{P}_2(q_2) = m_1 g h_1 + m_2 g h_2, \tag{3.6}$$

where $v_i$ is the linear velocity of the mass for link $i$, $h_i$ is the height of the $i$ link mass. To calculate the total kinetic energy (eq. 3.5) we need to calculate the squared velocities $v_1^2$ and $v_2^2$

$$v_1^2 = \dot{x}_1^2 + \dot{y}_1^2 = \left(\dot{q}_1 \frac{l_1}{2} \sin q_1\right)^2 + \left(\dot{q}_1 \frac{l_1}{2} \cos q_1\right)^2 = \dot{q}_1^2 \frac{l_1^2}{4}.$$

$$v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 = \left(l_1^2 + l_1 l_2 \cos q_2 + \frac{l_2^2}{4}\right)\dot{q}_1^2 + \left(l_1 l_2 \cos q_2 + \frac{l_2^2}{4}\right)\dot{q}_1 \dot{q}_2 + \frac{l_2^2}{4}\dot{q}_2^2.$$

[3] We do not call it end-effector's velocity since here we consider a single specific point on it: the end-effector could move to keep the point fixed

[4] Or space that is called Manifold, that is completely different structure from a vector space

Put (eq. 3.5) and (eq. 3.6) into (eq. 3.4) to get Lagrangian $\mathcal{L}$

$$\mathcal{L}(q,\dot{q}) = \mathcal{K}_1(q_1,\dot{q}_1) + \mathcal{K}_2(q_2,\dot{q}_2) - \mathcal{P}_1(q_1) - \mathcal{P}_2(q_2),$$

where the kinetic and potential energy terms are

$$\mathcal{K}_1(q_1,\dot{q}_1) = \frac{m_1\dot{q}_1^2 l_1^2}{8},$$
$$\mathcal{K}_2(q_2,\dot{q}_2) = \frac{m_2}{2}\Big[\Big(l_1^2 + l_1 l_2\cos q_2 + \frac{l_2^2}{4}\Big)\dot{q}_1^2 + \Big(l_1 l_2 \cos q_2 + \frac{l_2^2}{4}\Big)\dot{q}_1\dot{q}_2 + \frac{l_2^2}{4}\dot{q}_2^2\Big],$$
$$\mathcal{P}_1(q_1) = m_1 g\frac{l_1}{2}\sin q_1,$$
$$\mathcal{P}_2(q_2) = m_2 g\Big(l_1\sin q_1 + \frac{l_2}{2}\sin(q_1+q_2)\Big).$$

The equation of motion in terms of the Lagrangian function result from the following Euler Lagrange equation which are a necessary condition for Hamilton's principle of least action.

$$\frac{d}{dt}\Big(\frac{\partial\mathcal{L}}{\partial\dot{q}}\Big) - \frac{\partial\mathcal{L}}{\partial q} = \tau^\top, \tag{3.7}$$

where $\tau$ is the co-vector of the applied torques $\tau = \begin{pmatrix}\tau_1 & \tau_2\end{pmatrix}$. The equation of motion has to be derived for both generalized coordinates

$$\frac{\partial\mathcal{L}}{\partial q_1} = -m_1 g\frac{l_1}{2}\cos q_1 - m_2 g l_1\cos q_1 - m_2 g\frac{l_2}{2} - \cos(q_1+q_2),$$
$$\frac{\partial\mathcal{L}}{\partial\dot{q}_1} = \frac{1}{4}m_1 l_1^2\dot{q}_1 + m_2\Big(l_1^2 + \frac{l_2^2}{4}\Big)\dot{q}_1 + \Big(m_2 l_1 l_2\cos q_2\Big)\dot{q}_1 + \Big(\frac{1}{2}m_2 l_1 l_2\cos q_2\Big)\dot{q}_2 + \Big(\frac{1}{4}m_2 l_2^2\Big)\dot{q}_2,$$
$$\frac{\partial\mathcal{L}}{\partial q_2} = -\Big(\frac{1}{2}m_2 l_1 l_2\sin q_2\Big)\dot{q}_1^2 - \Big(\frac{1}{2}m_2 l_1 l_2\sin q_2\Big)\dot{q}_1\dot{q}_2 - \Big(\frac{1}{2}m_2 l_2\cos(q_1+q_2)\Big)g,$$
$$\frac{\partial\mathcal{L}}{\partial\dot{q}_2} = \Big(\frac{1}{2}m_2 l_1 l_2\cos q_2\Big)\dot{q}_1 + \frac{1}{4}m_2 l_2^2\dot{q}_1 + \frac{1}{4}m_2 l_2^2\dot{q}_2.$$

By taking the full time derivative and assembling into the equation 3.7 we get the expressions for $\tau_1$ and $\tau_2$

$$\tau_1 = \Big(\frac{1}{4}m_1 l_1^2 + m_2\big(l_1^2 + l_1 l_2\cos q_2 + \frac{1}{4}l_2^2\big)\Big)\ddot{q}_1 + \frac{1}{2}m_2 l_2\Big(l_1\cos q_2 + \frac{1}{2}l_2\Big)\ddot{q}_2$$
$$-\frac{1}{2}m_2 l_1 l_2\sin q_2\Big(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2\Big) + \Big(\frac{1}{2}m_1 + m_2\Big)g l_1\cos q_1 + \frac{1}{2}m_2 g l_2\cos(q_1+q_2),$$
$$\tau_2 = \Big(\frac{1}{2}m_2 l_2(l_1\cos q_2 + \frac{1}{2}l_2)\Big)\ddot{q}_1 + \frac{1}{2}m_2 l_2^2\ddot{q}_2 + \frac{1}{2}m_2 l_1 l_2\sin q_2\dot{q}_1^2 + \frac{1}{2}m_2 g l_2\cos(q_1+q_2).$$

## Equation of motion

The complicated expressions for $\tau_1$ and $\tau_2$ can be nicely assembled into the matrix equation of motion

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + N(q) = \tau^\top, \tag{3.8}$$

where $q, \dot{q}, \ddot{q}$ are vectors of generalized joints positions, angular velocity and angular acceleration respectively, $M(q)$ is called the inertia matrix, $C(q,\dot{q})$ is called unprecisely [15] the vector of Coriolis and centrifugal forces, $N(q)$ is the vector of potential forces

$$M(q) = \begin{pmatrix} 0.25m_1 l_1^2 + m_2\iota & 0.5m_2 l_2(\xi + 0.5l_2) \\ 0.5m_2 l_2(\xi+0.5l_2) & 0.5m_2 l_2^2, \end{pmatrix},$$
$$C(q,\dot{q})\dot{q} = \begin{pmatrix} -0.5m_2 l_1 l_2\sin q_2(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2) \\ 0.5m_2 l_1 l_2\sin q_2\dot{q}_1^2, \end{pmatrix},$$
$$N(q) = \begin{pmatrix} (0.5m_1 + m_2)g l_1\cos q_1 + 0.5m_2 g l_2\cos(q_1+q_2) \\ 0.5m_2 g l_2\cos(q_1+q_2), \end{pmatrix},$$

where $\xi = l_1\cos q_2$, $\iota = l_1^2 + l_2\xi + 0.25l_2^2$, $g = 9.80665\ m/s^2$. The detailed description of the physical meaneing and their properties are presented later on.

The 2R robot (Fig. 3.1) with $\tau = 0$ behaves like a double pendulum. In this subsection, we derived the equation of motion for the robot. Now we can use the derived mathematical model to synthesis control signals to get steer a robot in arbitrary direction.

## 3.3 2R robot control

The purpose of control is to steer a robot to follow a desired behavior. The robot's end-effector has to travel along the desired trajectory and with the desired force bandwidth. The position control does not suit for a robot that physically interacts with the environment. The impedance method of control is commonly used to handle an open-chain robot's dynamic physical interaction with the environment [16].

The left side of the equation 3.8 is a model of a robot and our task is to synthesis control signal at the right side such that we will be able to steer a robot in arbitrary direction. Thus, the total torque applied to the robot's links is a sum of three components:

$$\tau^\top = (\tau_g + \tau_f + \tau_i)^\top, \tag{3.9}$$

where $\tau_g$ is torque applied by actuators in the joints to compensate gravitation, $\tau_f$ is torque to compensate friction, and $\tau_i$ is equivalent torque consider dynamic interaction with the environment. To compensate robot's gravity it is enough to apply the vector of potential forces

$$\tau_g^\top = N(q) = \frac{\partial V}{\partial q}(q).$$

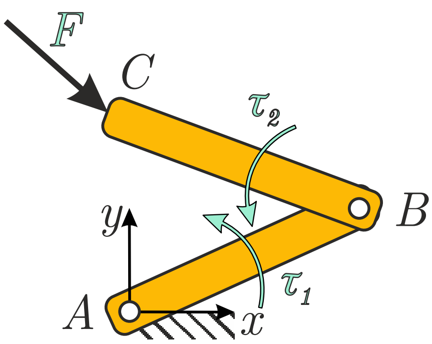



Figure 3.2: External force $F$ applied to the 2R robot

Fig. 3.2 shows the 2R robot interacting with the environment (Fig. 3.2). Force $F$ is applied to the robot's point $C$. In order to hold the equilibrium position, actuators have to produce reaction torques with equal power

$$\mathcal{P}_{joints} = \tau_i \dot{q} = F v_C = \mathcal{P}_{end-effector}, \tag{3.10}$$

where $F = (f_x \; f_y)$ is a co-vector of external force, $v^\top = (v_x \; v_y)$ is a velocity vector, $\tau = (\tau_1 \; \tau_2)$ is a co-vector of toques at joints, and $\dot{q}^\top = (\dot{q}_1 \; \dot{q}_2)$ is a links' angular velocity, and $\mathcal{P}$ is a scalar power. To get the expression for $\tau_i^\top$ we need to put eq. 3.2 into eq. 3.10 and transpose it:

$$\begin{gathered} FJ\dot{q} = \tau_i \dot{q}, \\ \tau_i = FJ, \\ \tau_i^\top = J^\top F^\top, \end{gathered}$$

where $J^\top$ is the transposed Jacobian matrix. To get the impedance behavior, the robot's end-effector $C$ is attached to a point of a desired trajectory via a virtual spring. An elastic force $F$ is applied to the end-effector

$$F^\top = P(\tilde{p} - p) + D(\tilde{v} - v), \tag{3.11}$$

where $P$ and $D$ are proportional (or elastic) and derivative (or damping) coefficients, $\tilde{p}$ and $p$ are virtual and current position vector of the end-effector, $\tilde{v}$ and $v$ are desired and current velocity vector of the end-effector respectively. The desired position can be stated as

$$\tilde{p} = \begin{pmatrix} A_x \cos(ft) + B_x \\ A_y \sin(ft) + B_y \end{pmatrix}, \tag{3.12}$$

where $A_x$ and $A_y$ are amplitudes and $B_x$ and $B_y$ are biases for $\hat{x}$ and $\hat{y}$ respectively, $f$ is frequency of oscillation, $t$ is time. Impedance control helps to create a virtually passive system that can dynamically interact with the environment.

Forces generated by friction are varying but always present. They are in general a function of $q$ and $\ddot{q}$. A simple, position dependent, linear model of the viscous component is multiplication of friction (damping) function $b(q)$ and angular velocity $\dot{q}$

$$\tau_f^\top = -b(q)\dot{q}.$$

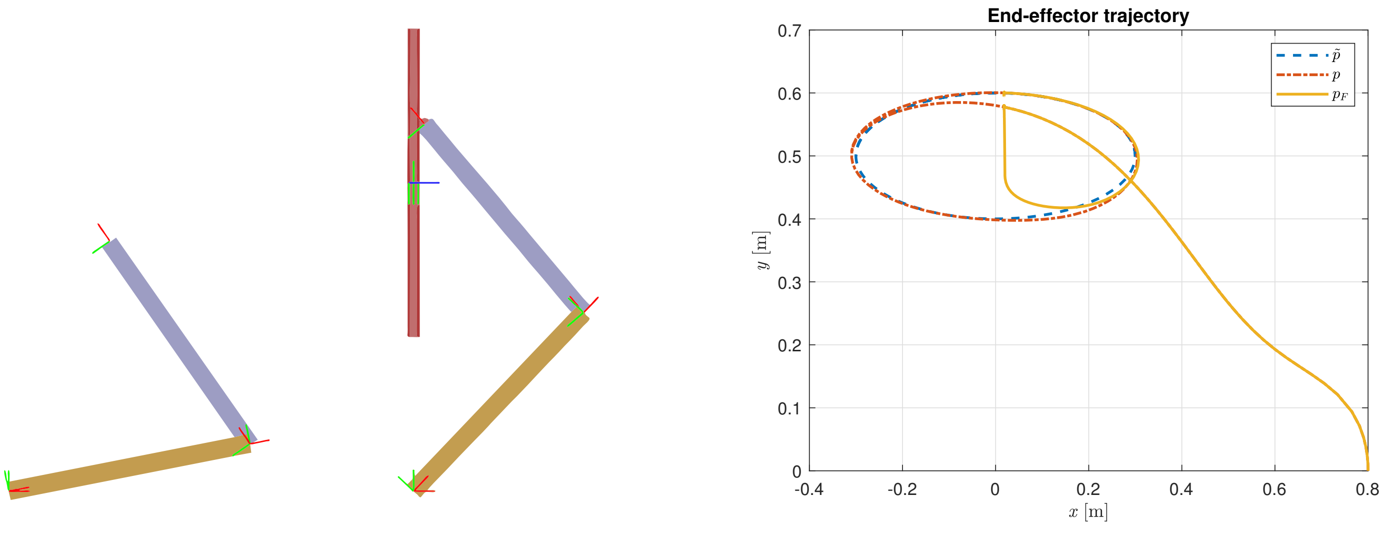
Equation (3.9) calculates separate total control signals to be applied for the both links.

## 3.4 2R robot simulation

A "*virtual experiment*" is a useful tool to check designs and derive control algorithms of robots. A simulation is a good tool to evaluate and support fundamental design choices, study interaction between different parts of a complicated mechanical structure and environment, and further plan reference trajectories and tune controllers [17]. Since a mechatronic system is about interaction between mechanics, electronics, and information, simulation should be performed with respect to all domains. In order to do that, we have implemented a simulation in MATLAB Simscape Multibody using Contact Forces Library.

In order to model the physical interaction of a planar 2R collaborative robot with an external object, we have implemented a penalty force approach, which allows a small overlap of the bodies using the MATLAB's Contact Library.

Table 3.1 shows all parameters for a 2R robot: physical properties, desired trajectory coefficients for eq. 3.12 and applied force of eq. 3.11. Fig. 3.3 shows a visualization of the 2R robot without (Fig. 3.3a) and with external object (Fig. 3.3b), and the simulation results (Fig. 3.3c). The blue dashed line shows the desired trajectory $\tilde{p}$ according to eq. 3.12 and parameters in Table 3.1. The red dash-and-dot line shows the actual trajectory $p$ of the end-effector in the simulation without any physical contact with the environment. The red line travels from the initial position of the robot to the desired trajectory. The solid yellow line shows the actual trajectory $p_F$ of the end-effector in the simulation with contact modeling of physical interaction with an external object. The orange line is coincident with the red one before and after the contact, while the contact end-effector travels along the surface.




(a) (b) (c)

Figure 3.3: Simulation visualization of 2R robot without **(a)** and with external object **(b)**, and trajectories of the end-effector **(c)**: $\tilde{p}$ is the desired trajectory, $p$ is the actual trajectory without contact modeling, and $p_F$ is the actual trajectory with contact

| Parameter | $m_1$ | $m_2$ | $l_1$ | $l_2$ | $A_x$ | $A_y$ | $f$ | $B_x$ | $B_y$ | $K$ | $D$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Value | 2 kg | 2 kg | 0.4 m | 0.4 m | 0.3 m | 0.1 m | $\pi$ | 0 m | 0.5 m | 600 | 100 |

Table 3.1: Parameters for 2R robot simulation

In this section, we have taken a look at a simple example of the design, modeling, and control of a 2R planar robot. However, the presented methods are not enough working with for complex systems. The general idea of mechanical analysis, modeling and control is the same, but we need the right tools to model spatial multibody systems. In the next section, we begin to study the the new tools for modeling spatial systems.

# Bibliography


[1] Stefano Stramigioli. *Modeling and IPC control of interactive mechanical systems—A coordinate-free approach*. 2001.

[2] Vincent Duindam and Stefano Stramigioli. *Modeling and control for efficient bipedal walking robots: A port-based approach*, volume 53. Springer, 2008.

[3] Vincent Duindam, Alessandro Macchelli, Stefano Stramigioli, and Herman Bruyninckx. *Modeling and control of complex physical systems: the port-Hamiltonian approach*. Springer Science & Business Media, 2009.

[4] Cristian Secchi, Stefano Stramigioli, and Cesare Fantuzzi. *Control of interactive robotic interfaces: A port-Hamiltonian approach*, volume 29. Springer Science & Business Media, 2007.

[5] Kevin M Lynch and Frank C Park. *Modern Robotics*. Cambridge University Press, 2017.

[6] Richard M Murray, Zexiang Li, S Shankar Sastry, and S Shankara Sastry. *A mathematical introduction to robotic manipulation*. CRC press, 1994.

[7] Antonio Tornambe and Claudio Melchiorri. *Modelling and Control of Mechanisms and Robots*. World Scientific, 1996.

[8] Jon M Selig. *Geometric fundamentals of robotics*. Springer Science & Business Media, 2004.

[9] Henry M Paynter. *Analysis and design of engineering systems*. MIT press, 1961.

[10] Darina Hroncová, Alexander Gmiterko, Peter Frankovskỳ, and Eva Dzurišová. Building elements of bond graphs. In *Applied Mechanics and Materials*, volume 816, pages 339–348. Trans Tech Publ, 2015.

[11] Catherine Bidard. Screw-vector bond graphs for multibody systems modelling. *SIMULATION SERIES*, 25:195–195, 1993.

[12] B Gola, J Kopec, J Rysiński, and S Zawislak. Bond graph model of a robot leg. In *Graph-Based Modelling in Engineering*, pages 69–80. Springer, 2017.

[13] Shengqi Jian, Cheng Yin, Luc Rolland, and Lesley James. Five bar planar manipulator simulation and analysis by bond graph. In *ASME International Mechanical Engineering Congress and Exposition*, volume 46476, page V04AT04A026. American Society of Mechanical Engineers, 2014.

[14] Zhaohong Wu, Matthew I Campbell, and Benito R Fernández. Bond graph based automated modeling for computer-aided design of dynamic systems. *Journal of Mechanical Design*, 130(4), 2008.

[15] S Stramigioli and PC Breedveld. An interpretation of the eulerian junction structure in 3d rigid bodies. *Iji*, 1:2, 1994.

[16] Neville Hogan. Impedance control: An approach to manipulation. In *1984 American control conference*, pages 304–313. IEEE, 1984.

[17] RA Hyde and J Wendlandt. Tool-supported mechatronic system design. In *2008 34th Annual Conference of IEEE Industrial Electronics*, pages 1674–1679. IEEE, 2008.