Chapter 7

Control

Generally, robots are devices to accomplish various tasks. The purpose of control is to achieve the desired behavior for a given robotic system, a given task, and a robot's environment. Robots can follow a predetermined trajectory, or they can act as s sources of forces, or they can dynamically interact with the environment and a human being.

It is common to classify the control strategies as *position control* (or motion control), *force control*, and *impedance control* (or interaction control). Each of these behaviors is appropriate depends on both the task and the environment. Position control can be used only if the robot moves freely in space and do not touch anything. The force control has to be used when the robot's end-effector is in contact with something and only force is changing. And it makes sense to use impedance control if there is a possibility to touch something while the robot is following a virtual position. For collaborative and wearable robots, impedance control is the most interesting since it couples forces and positions of robots with respect to interaction with a human.

Once the control strategy is chosen, we can use feedback control to measure position, velocity, or force by means of sensors to measure the current state of a robot, compare it with the desired behavior, and generate control signals to be applied to the robot. This section gives an overview of control strategies.

7.1 Control overview

Suppose a simple 1 degree of freedom robot with mass m connected with a fixed frame via a rotational joint. The sensors such as potentiometers or encoders can measure the robot's state q, e.g., joint position, and tachometers can be used to measure joint velocity $\dot{q}$. The controller receives the data from sensors and updates control signals to the actuators to apply a control force τ. Figure 7.1 shows a diagram of the interaction for an abstract robotic device with an environment or a person, which applies an external force F to a robot. The resulting behavior depends on both controlled and external force since there is a relation between external force F and the robot's state q.

The real physical system suffers from unknown flexibility and vibrations in joints, backlash in gears, actuators limits, and limited resolution sensors [5] that we do not intend to consider in this section.

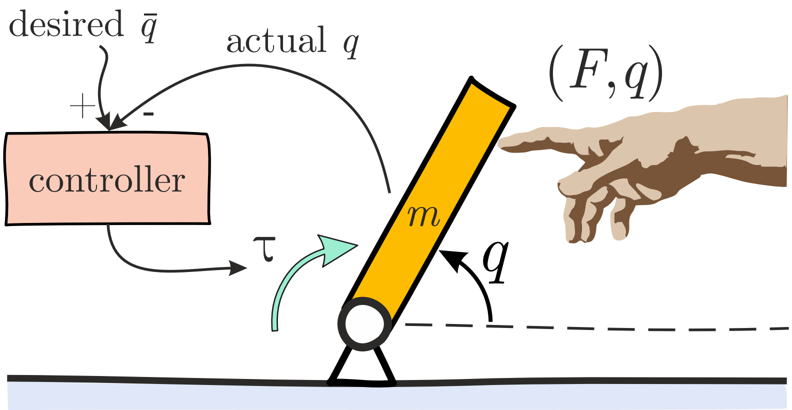

Figure 7.1: Interaction with a robot

To obtain the desired behavior for a robot the controller must take into account the internal state of the device q and external disturbances F, and set the control law τ. The environment will also have its own behavior. It makes sense to use *position control* when the robot is isolated which means the external force must be zero

$$F(t) = 0, \ \forall t,$$

and $q(t)$ only depends on robot's configuration.

Force control can be used in scenarios when an end-effector of a robot in contact with an object, i.e., it is "glued" to a fixed point, which means the end-effector's velocity must be zero

$$\dot{q}(t) = 0, \ \forall t,$$

and $F(t)$ only depends on robot's configuration.

In dynamic *interaction control*, both the force $F(t)$ and the position $q(t)$ depend on both the state of the robot and the state of the environment. That results in a unique solution for $F(t)$ and $q(t)$. We cannot control the force $F(t)$ and/or position $q(t)$ independently of the state of the environment, but we can control the state of the robot independently of the environment.

7.2 Position control

7.2.1 Defining of error dynamics

Let us consider control dynamics for a single joint depicted in Figure 7.1. The concept generalize easy for a more complex systems.

Suppose to have a desired joint trajectory $\bar{q}(t)$ we want to follow. If we define a error $e(t)$ to be

$$e(t) := \bar{q}(t) - q(t),$$

where $q(t)$ is the actual position. For tracking we want that

$$\lim_{t \to \infty} e(t) = 0.$$

The differential equation that describe evolution for a joint error $e(t)$ is called the *error dynamics*

$$\ddot{e} + K_v \dot{e} + K_p e = 0,$$

where K_p and K_v are positive definite proportional (elastic) coefficient and differentiation (damping) coefficient respectively. The purpose of the feedback controller is to create an error dynamics such as the differentiable equation goes to zero.

For second-order error dynamics, a good mechanical analogy is a linear mass-spring-damper system that can be described as the following equation

$$m\ddot{e} + b\dot{e} + ke = F,$$

where m is the mass of a body, that is connected with a fixed frame via a spring with a coefficient of stiffnesses k and damper with damping constant b; here e is a displacement of the body, i.e., the deflection of a spring; force F is applied to the body as an external force. The damper generates a force $-b\dot{e}$ and the spring applies a force $-ke$ to the mass.

7.2.2 Error dynamics for extremely simple robot

Let us consider an example with an extremely simple robot that can be described as a linear system with

$$M(q) = I, \ C(q, \dot{q}) = 0, \ G(q) = 0, \ F(q, \dot{q}) = 0, \ u = \tau_a^\top.$$

For such a system the dynamics equation (6.1) will be the following

$$\ddot{q} = u.$$

The task of controlling such a system is easy. The desired error dynamics for the system $\ddot{q} = u$ can be achieved with the feedback controller

$$u = \ddot{q} = \ddot{q}_d + K_v \dot{q}_d + K_p q_d - (K_v \dot{q} + K_p q),$$

where $\ddot{q}_d$, $\dot{q}_d$, q_d is specified by a user, $\dot{q}$, q is measured, and $\ddot{q}$ is the acceleration, i.e., feedback law. For the considered simple robot, if we apply the control law the system will be follow the *error dynamics*.

7.2.3 Error dynamics for general robot

However, generally we are dealing with more complex systems. For such systems we can create a model of a robot, compute a torque, and do feedback linearizion. We can consider a model of a general system

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(q,\dot{q}) + G(q) = \tau_a^\top. \tag{7.1}$$

If the matrix components are known and it is possible to measure $\dot{q}$ and q, we can apply the following control law with a new input u

$$\tau_a^\top = M(q)u + C(q,\dot{q})\dot{q} + F(q,\dot{q}) + G(q). \tag{7.2}$$

If we simplify the terms in eq. (7.1) and eq. (7.2) we would get

$$M(q)\ddot{q} = M(q)u \quad \underbrace{\implies}_{M(q) \text{ is pos. def.}} \quad \ddot{q} = u. \tag{7.3}$$

Thus, we have created a virtual system with known simple dynamics, and we can use the same *error dynamics*.

However, a model *always* differs comparing with a real physical system. Because of that, our model will always consist of estimations

$$\hat{M}(q), \hat{C}(q,\dot{q}), \hat{F}(q,\dot{q}), \hat{V}(q).$$

In the control law only the terms with ˆ take place, the terms without hats are physical quantities, and they are measured with some level of accuracy. Considering the estimations, we can get the following control equation

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(q,\dot{q}) + V(q) = \hat{M}(q)u + \hat{C}(q,\dot{q})\dot{q} + \hat{F}(q,\dot{q}) + \hat{V}(q),$$

$$0 = \hat{M}(q)u - M(q)\ddot{q} + \hat{C}(q,\dot{q})\dot{q} - C(q,\dot{q})\dot{q} + \hat{F}(q,\dot{q}) - F(q,\dot{q}) + \hat{V}(q) - V(q),$$

$$M(q)\ddot{q} = \hat{M}(q)u + \hat{C}(q,\dot{q})\dot{q} - C(q,\dot{q})\dot{q} + \hat{F}(q,\dot{q}) - F(q,\dot{q}) + \hat{V}(q) - V(q),$$

Instead of eq. (7.3) we can get the following control law

$$\ddot{q} = \Delta M u + M^{-1}(q)(\Delta C\dot{q} + \Delta F + \Delta V),$$

where $\Delta M = M^{-1}(q)\hat{M}(q)$, $\Delta C = \hat{C}(q,\dot{q}) - C(q,\dot{q})$, $\Delta F = \hat{F}(q,\dot{q}) - F(q,\dot{q})$, $\Delta V = \hat{V}(q) - V(q)$. If our estimation is extremely correct the ΔM will be equal to identity matrix I, and all others terms will equal zeros matrices, and the eq. (7.3) will hold.

In this section, we have briefly introduced position control using computed torque by means of feedback linearization and considered its robustness problem. However, generally, it makes sense to use control strategies that aim for a position error $e \to 0$ as $t \to \infty$ for isolated robotic systems that do not interact with the environment.

7.3 Force control

Force control can be used for interaction tasks with the environment. However, that force control can only be used if the task is not to create motion at the end-effector but to apply forces and torques to the environment. The force control only can be used if there is a contact between a robot and the environment. Usually, force control is applied in some direction, while in another direction, a robot can move freely.

We can consider the dynamic effect of a wrench $W^{0,n}$ applied to the end-effector with a fixed frame Ψ_n expressed in frame Ψ_0. The power P_{ee} supplied to the robot by the external wrench

$$P_{ee} = W^{0,n}T_n^{0,0},$$

where $T_n^{0,0}$ is a Twist of the end effector n, with respect to the frame Ψ_0 and expressed in the same frame Ψ_0. We can calculate $T_n^{0,0}$ using the geometrical Jacobian $J(q)$ and velocity vector for all the robot's joints

$$T_n^{0,0} = J(q)\dot{q}.$$

It yields that the supplied power

$$P_{ee} = W^{0,n}J(q)\dot{q} = (J^\top(q)(W^{0,n})^\top)^\top\dot{q}.$$

We have to find a torque τ which need to be applied in robot's joints to compensate the dynamic effect of the wrench $W^{0,n}$. The power P_j that has to be generated in the joints

$$P_j = \tau\dot{q}.$$

The powers must be equal $P_{ee} = P_j$ to compensate the dynamic effect of the external force

$$\tau\dot{q} = (J^\top(q)(W^{0,n})^\top)^\top \dot{q},$$

that yields in equation for torques that must be generated by joints

$$\tau^\top = J^\top(q)(W^{0,n})^\top. \tag{7.4}$$

The manipulator equation of dynamics 6.1 can be written as

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + F(q,\dot{q}) + G(q) + J^\top(q)(W^{0,n})^\top = \tau^\top.$$

A robot controlled by the force control typically do not move or moves slowly during a force control task. Thus, we can ignore the acceleration and velocity terms

$$G(q) + J^\top(q)(W^{0,n})^\top = \tau^\top.$$

Joint angular position feedback can be used to implement the Force control law

$$\tau^\top = \hat{G}(q) + J^\top(q)(\bar{W}^{0,n})^\top,$$

where $\hat{G}(q)$ is a model of the gravitational torques and $\bar{W}^{0,n}$ is the desired wrench. The Force control law needs a precision model for gravity compensation and precise position sensors allocated with actuators.

It can be seen that the Force control strategy is not the best option for dynamic robot-human interaction.

7.4 Interaction control

The impedance control is an approach to manipulation was introduced in [16], after realizing that position or force control is inadequate control strategies for dynamic interaction tasks. For the interaction tasks a robot can not be treated as isolated system.

To implement the impedance control for a robot we need to attach a virtual spring that connects the robot's end-effector Ψ_n and the virtual frame Ψ_v

$$H_n^0(q) \to H_v^0,$$

where H_v^0 represents the virtual position that is needed to complete the task. The virtual trajectory can be treated as an array of frames Ψ_v. The force generated by the virtual spring is a function of the relative position between Ψ_n and Ψ_v

$$H_n^v = H_0^v H_n^0(q).$$

The spring's force can be consider as a Wrench $W_S^{0,n}(H_v^n)$

$$W_S^{0,n}(H_v^n) = \begin{pmatrix}\tau^n & f^n\end{pmatrix},$$

where τ^n is a torque and f^n is the force generated by a spatial virtual spring. A way to implement a model of spatial spring is presented in [18]. The impedance control utilizing the concept with spatial virtual spring is given in [19] and explained in detail in [1]. The torque and force generated by the spatial virtual spring can be calculated as

$$\tilde{\tau}^n = -2\mathrm{as}(G_o R_v^n) - \mathrm{as}(G_t R_n^v \tilde{p}_v^n \tilde{p}_v^n R_v^n) - 2\mathrm{as}(G_c \tilde{p}_v^n R_v^n),$$

$$\tilde{f}^n = -R_n^v \mathrm{as}(G_t \tilde{p}_v^n) R_v^n - \mathrm{as}(G_t R_n^v \tilde{p}_v^n R_v^n) - 2\mathrm{as}(G_c R_v^n),$$

where as() is an operator which takes the skew-symmetric part of a matrix, R_v^n and p_v^n are the subparts of the matrix

$$H_v^n = \begin{pmatrix} R_v^n & p_v^n \\ 0 & 1 \end{pmatrix}$$

and G_o, G_t, G_c are called respectively orientational, translational and coupling co-stiffnesses of the spring [19].

The Wrench $W_S^{0,n}(H_v^n)$ can be used to calculate the torque τ_i that need to be applied in the joints to compensate the external load (eq. 7.4)

$$\tau_i^\top = J^\top(q)(W_S^{0,n}(H_v^n))^\top, \tag{7.5}$$

where subscript i stands for *interaction* or *impedance*.

The control law would be

$$\tau_d + \tau_i + \tau_f = \hat{M}(q)\ddot{q} + \hat{C}(q,\dot{q})\dot{q} + \hat{F}(q,\dot{q}) + \hat{G}(q) + J^\top(q)(W_S^{0,n}(H_v^n))^\top - B(q)\dot{q}$$

where torques components τ_d, τ_i, and τ_f stands for gravity compensation, interaction and friction respectively.

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.

[18] Stefano Stramigioli and Vincent Duindam. Variable spatial springs for robot control applications. In *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180)*, volume 4, pages 1906–1911. IEEE, 2001.

[19] Ernest D Fasse and Jan F Broenink. A spatial impedance controller for robotic manipulation. *IEEE Transactions on Robotics and Automation*, 13(4):546–556, 1997.