Modelling and Control of the Dynamical Mechanical Systems and Walking Robots

Sergej Čelikovský

Department of Control Engineering
Faculty of Electrical Engineering
Czech Technical University in Prague

and

Institute of Information Theory and Automation
Czech Academy of Sciences

Outline

Walking robots
Classification of walking robots

Static walking:
center of mass always above the feet, the robot can not fall, even if stopped at any time.

Dynamic walking: center of mass not always above the feet, it should go on moving not to fall

Two-dimensional walking (2D-walking):
Only sagittal plane studied, the forward movement and stability is believed to be the crucial for the walking-like movement.

Three-dimensional walking (3D-walking):
Full orientation including the lateral movement studied, the lateral balance should be (even intuitively) easier to handle, than the forward movement.

Walking robots

Classification of walking robots

Fully actuated walking robots:

- Large feet in full contact with the ground and actuated ankles.
- Stable (static) walking usually fully actuated.
- Unstable (dynamic) walking can be also fully actuated.
- Zero moment point (ZMP) should be computed and ensured to be bellow the feet.

Underactuated walking robots:

- Unstable (dynamic) walking only.
- Underactuated angle is at the pivot point.
- Feet are absent or very small with weakly actuated ankles.

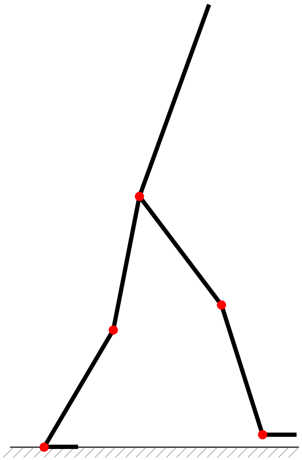
Walking robots

Fully actuated walking

- Slow movement, weak coupling between links and strong actuators - the kinematic trajectory planning possible and implementable through the standard control engineering design (*e.g.* PD or PID controllers). ZMP condidion impose actuators limitations.
- Fully actuated mechanical system is theoretically well understood even if its full dynamics (including forces and torques) should be considered - **computed torque principle**.
- The typical fully actuated static walking humanoids (like HONDA) are heavy, with strong joints actuation and very slow dynamic walking coupling between the links dynamics, or even with the static walking only.

Walking robot with knees, ankles, feet and torso

Fully actuated state - ZMP bellow the foot



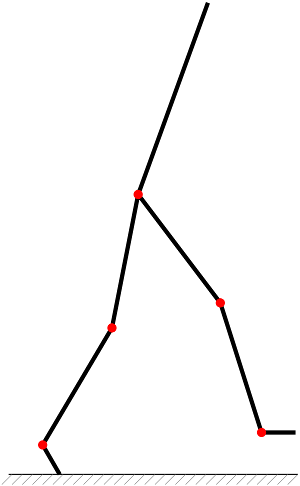
Walking robots

Intrinsic walking is the underactuated walking

- **BUT!** What if the walk should be swift, legs light, energy efficiency strived for? Then the fully actuated walking robotics is not an option.
- The ZMP condition is demanding and therefore the torque in ankles should be, anyway, small, only the balancing one and to facilitate the full foot contact with the ground.
- If the ZMP condition is violated, the full flat contact of the foot is lost and another angle - another degree of freedom appears - yet, there is unactuated angle again and the whole robot becomes underactuated.

Walking robot with knees, ankles, feet and torso

Underactuated situation - ZMP not bellow the foot



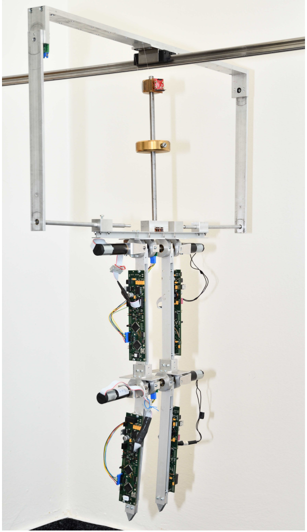
Walking robots

Intrinsic walking is the underactuated walking

- Real walking: both the fully actuated and the underactuated walking phases present.
- **Conclusion:** It is reasonable to study the underactuated walking, as a natural abstraction of weak ankles up to the case with NO ankles.

Walking robots

Planar walking



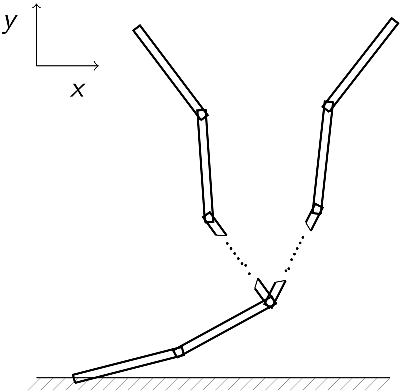
Walking robots

Plan: The **underactuated planar walking** will be further studied in detail.

Indeed, as noted before, the control along the sagittal plane is the most challenging and interesting one. Instability is actually the source of the movement forward.

Underactuated mechanical systems
Introduction

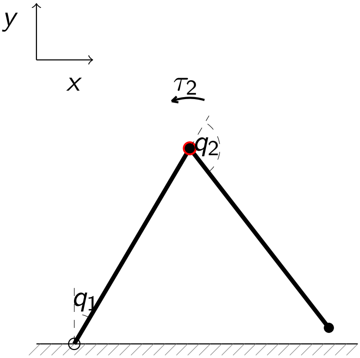
- Less actuators than **degrees of freedom (DOF]**
- The n-link with $n-1$ actuators
- Nonlinear techniques needed
- Pendubot
- Acrobot
- Rotational inverted pendulum
- ...



Underactuated mechanical systems

Acrobot (aka Compass-Gait Walker (CGW))

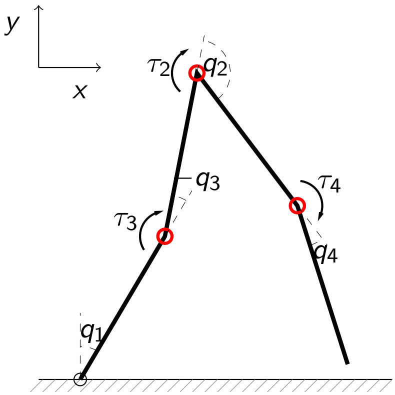
- The simplest underactuated walking model, 2 DOF, 1 actuator.
- To model walking, the underactuated angle is at the pivot point.
- "Planar" walking-like movement possible only theoretically.
- **Acrobot**, or **Compass-Gait Walker (CGW)**, ...



Underactuated mechanical systems

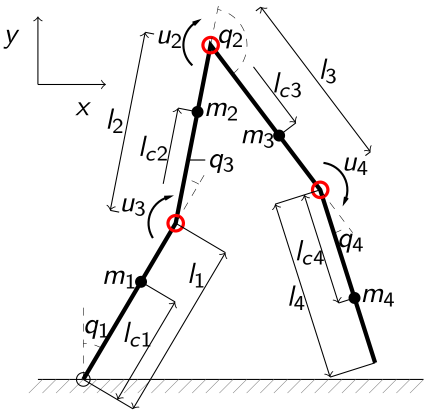
The four-link walker

- Acrobot walking: unrealistic
- 4-link: more realistic
- 4-link: 4 DOF, 3 actuators
- Legs with knees, without feet



Underactuated mechanical systems

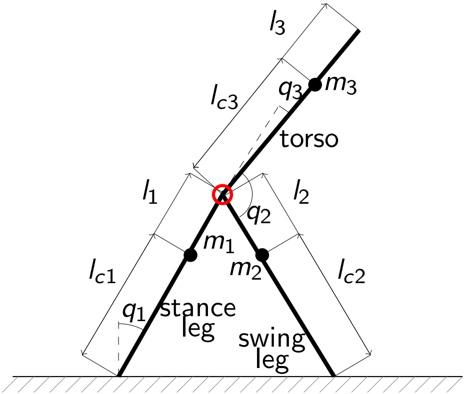
The four-link walker - Configuration and physical parameters



m_1, m_4	1 [Kg]	m_2, m_3	1.5 [Kg]
l_1, l_4	0.5 [m]	l_{c1}, l_{c4}	0.3 [m]
l_2, l_3	0.6 [m]	l_{c2}, l_{c3}	0.4 [m]

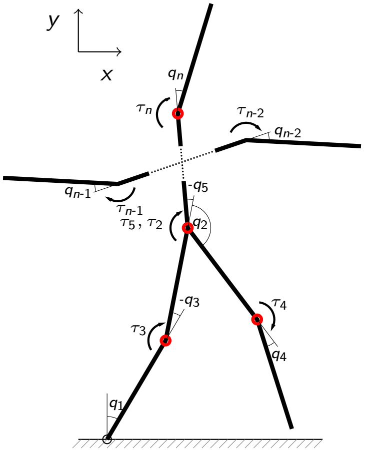
Underactuated mechanical systems

The three-link (aka Compass-Gait Walker with Torso)



Underactuated mechanical systems

Possible general underactuated planar *n*-link walker scheme



Modeling of the walking robots

Two different model phases

1. Continuous-time phase (aka single support phase, ...). Model: ordinary differential equations with control inputs, resulting in the standard controlled continuous-time system studied in various subjects before (B3B35ARI, B3M35NES, B3M35LSY).

2. Discrete-time phase (aka double-support phase). Model: uncontrolled mappings of the state to the new state, with some jump, or also impulsive system. It is a single application of a discrete-time system without input.

The continuous-time models of the mechanical systems

Euler-Lagrange formalism

- Choose the generalized coordinates $q = (q_1, \ldots, q_n)^\top$ and the generalized velocities $\dot{q} = (\dot{q}_1, \ldots, \dot{q}_n)^\top$.
- Compute the system kinetic energy $\mathcal{K}(q, \dot{q})$, potential energy $\mathcal{V}(q)$ and Lagrangian $\mathcal{L}(q, \dot{q})$:

$$\mathcal{L}(q, \dot{q}) = \mathcal{K} - \mathcal{V} = \frac{1}{2}\dot{q}^T D(q)\dot{q} - \mathcal{V}(q).$$

- System dynamics given by the Euler-Lagrange formalism

$$\begin{bmatrix} \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_1} - \frac{\partial \mathcal{L}}{\partial q_1} \\ \vdots \\ \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_n} - \frac{\partial \mathcal{L}}{\partial q_n} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_n \end{bmatrix},$$

- $\tau_1, \ldots, \tau_n$ are generalized external forces acting **along** the generalized coordinates $q_1, \ldots, q_n$, respectively.
- $D(q) = D(q)^\top > 0$ is the **inertia (aka mass) matrix**.

The continuous-time models of the mechanical systems

Euler-Lagrange formalism

- The Euler-Lagrange formalism gives the system of the second-order ordinary differential equations

$$D(q)\ddot{q}+C(q,\dot{q})\dot{q}+G(q) = (\tau_1,\ldots,\tau_n)^\top,\ G(q) = -\left[\frac{\partial \mathcal{V}}{\partial q}\right]^\top,$$

$$C(q,\dot{q})\dot{q} = \left[\sum_{i=1}^{n} \frac{\partial D(q)}{\partial q_i}\dot{q}_i\right]\dot{q} - \left[\frac{\partial}{\partial q}\left[\frac{1}{2}\dot{q}^T D(q)\dot{q}\right]\right]^\top,$$

  $C(q,\dot{q})$ is the matrix of the Coriolis and centrifugal forces, $G(q)$ is the gravity vector.

- The choices of generalized coordinates and generalized velocities are related: $\sum_{i=1}^n \tau_i \mathrm{d}q_i$ should be the infinitesimal increment of energy of the system done by work of external forces. In particular, if generalized coordinates are angles, then the generalized forces are **torques**.

The continuous-time models of the mechanical systems

Fully actuated controlled system in standard form of the first-order ODE

- Controlled inputs (actuators) are the generalized forces, all of them available and independent - **full actuation**:

$$u = \begin{bmatrix} u_1 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_n \end{bmatrix}$$

- The state vector is $x = (x_1, \ldots, x_{2n})^\top = (x^1, x^2)^\top$

$$x^1 = (q_1, \ldots, q_n),\; x^2 = (\dot{q}_1, \ldots, \dot{q}_n).$$

- The standard form of the first-order controlled system:

$$\dot{x} = f^{ST}(x) + G^{ST}(x)u, \quad G^{ST}(x) = D(x^1)^{-1}$$

$$f^{ST}(x) = \left(x^2, \overline{f}(x)\right)^\top,\; \overline{f}(x) = D^{-1}(x^1)\left(-C(x^1, x^2)x^2 - G(x^1)\right)^\top$$

The continuous-time models of the mechanical systems

Exact feedback feedback linearization and computed torque principle (inverse dynamics)

For fully actuated system take the feedback (introduce the new "virtual" input $\overline{u}$)

$$\overline{u} = \overline{f}(x) + G^{ST}(x)u, \quad u = [G^{ST}(x)]^{-1}(\overline{u} - \overline{f}(x)),$$

which gives the simple linear system of n double integrators

$$\dot{x}_1 = x_{n+1}, \ldots, \dot{x}_n = x_{2n}, \ \dot{x}_{n+1} = \overline{u}_1, \ldots, \dot{x}_{2n} = \overline{u}_n.$$

This gives the simple **model based** implementation of any smooth kinematically planned trajectory $q^r(t)$, containing some PD controller (but any gains $k_i^p < 0, \ k_i^d < 0, i = 1, \ldots, n$, sufficient):

$$u = [G^{ST}(x)]^{-1} \left[\begin{pmatrix} \ddot{q}_1^r + k_1^p(q_1 - q_1^r) + k_1^d(\dot{q}_1 - \dot{q}_1^r) \\ \vdots \\ \ddot{q}_n^r + k_n^p(q_n - q_n^r) + k_n^d(\dot{q}_n - \dot{q}_n^r) \end{pmatrix} - \overline{f}(x) \right].$$

The continuous-time models of the mechanical systems

Exact feedback feedback linearization and computed torque principle (inverse dynamics)

Indeed, for virtual input $\overline{u}$ that is equivalent to

$$\overline{u} = \begin{pmatrix} \ddot{q}_1^r + k_1^p(q_1 - q_1^r) + k_1^d(\dot{q}_1 - \dot{q}_1^r) \\ \vdots \\ \ddot{q}_n^r + k_n^p(q_n - q_n^r) + k_n^d(\dot{q}_n - \dot{q}_n^r) \end{pmatrix}$$

Recalling, that $x^1 = (q_1, \ldots, q_n),\ x^2 = (\dot{q}_1, \ldots, \dot{q}_n)$, $\dot{x}_1 = x_{n+1}, \ldots, \dot{x}_n = x_{2n},\ \dot{x}_{n+1} = \overline{u}_1, \ldots, \dot{x}_{2n} = \overline{u}_n$ and introducing

$$e_1 = q_1 - q_1^r, \ldots, e_n = q_n - q_n^r$$

gives

$$\dot{e}_1 = e_{n+1}, \ldots, \dot{e}_n = e_{2n},$$

$$\dot{e}_{n+1} = k_1^p e_1 + k_1^d e_{n+1}, \ldots, \dot{e}_{2n} = k_n^p e_n + k_n^d e_{2n}.$$

The continuous-time models of the mechanical systems

Exact feedback feedback linearization and computed torque principle (inverse dynamics)

Since $G^{ST}(x) = D(x^1)^{-1}$, $[G^{ST}(x)]^{-1} = D(x^1) = D(q)$, one has

$$[G^{ST}(x)]^{-1}\overline{f}(x) = C(q,\dot{q})\dot{q} + G(q)$$

$$u = D(q)\begin{pmatrix} \ddot{q}_1^r + k_1^p(q_1 - q_1^r) + k_1^d(\dot{q}_1 - \dot{q}_1^r) \\ \vdots \\ \ddot{q}_n^r + k_n^p(q_n - q_n^r) + k_n^d(\dot{q}_n - \dot{q}_n^r) \end{pmatrix} + C(q,\dot{q})\dot{q} + G(q).$$

COMPUTED TORQUE PRINCIPLE (CTP): the required torque is computed by substituting the desired linear second order dynamics of q into the second order Euler-Lagrange robot dynamics $u = D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q)$.

CTP introduced in robotics earlier than the exact feedback linearization (EFL) in nonlinear control theory.

Clearly, for the fully actuated mechanical systems EFL=CTP.

The continuous-time models of the mechanical systems

Underactuated mechanical systems

- Underactuated system dynamics given by the Euler-Lagrange formalism

$$\begin{bmatrix} \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_1} - \frac{\partial \mathcal{L}}{\partial q_1} \\ \vdots \\ \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_n} - \frac{\partial \mathcal{L}}{\partial q_n} \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \vdots \\ \tau_n \end{bmatrix}, \quad \tau_1 = \cdots = \tau_k = 0.$$

- Coordinates $q_1, \ldots, q_k$ **directly unactuated**, $q_{k+1}, \ldots, q_n$ **directly actuated**, k is the **underactuation degree**.
- Analogously, as for the fully actuated systems (D, C, G the same), the second order dynamics obtained.

$$D(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = (0, \ldots, 0, \tau_{k+1}, \ldots, \tau_n)^\top$$

The continuous-time models of the mechanical systems

Underactuated controlled system in standard form of the first-order ODE

$$
u = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ u_{k+1} \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \tau_{k+1} \\ \vdots \\ \tau_n \end{bmatrix}, \quad x = (x_1, \ldots, x_{2n})^\top = (x^1, x^2)^\top
$$

$$
x^1 = (q_1, \ldots, q_n), \; x^2 = (\dot{q}_1, \ldots, \dot{q}_n).
$$

$$
\dot{x} = f^{ST}(x) + G^{ST}(x)u, \quad G^{ST}(x) = D(x^1)^{-1}
$$

$$
f^{ST}(x) = \left(x^2, \overline{f}(x)\right)^\top, \; \overline{f}(x) = D^{-1}(x^1)\left(-C(x^1, x^2)x^2 - D(x^1)\right)^\top
$$

Exact feedback linearization and computed torque principle clearly not possible using the previously described approach.

The continuous-time models of the mechanical systems

Underactuated mechanical planar walking-like chains

- 2D-walking models have usually underactuation degree $k = 1$. The angle at the pivot point q_1 is unactuated, $q_2, \ldots, q_n$ directly actuated.
- For these planar walking-like chains, kinetic energy does not depend on q_1, *i.e.* $D(q) \equiv D(q_2, q_3, \ldots, q_n)$ and q_1 is called **cyclic variable**, $q_2, \ldots, q_n$ are called **shape variables**. Any **absolute orientation angle** is also cyclic variable. **Relative angles** are shape variables.
- 3D-walking models have underactuation degree $k = 2$, but only one cylic variable.

The continuous-time models of the mechanical systems

Underactuated mechanical planar walking-like chains

Summarizing, the underactuated planar walking models are as follows:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \begin{bmatrix} 0 \\ u_2 \\ \vdots \\ u_n \end{bmatrix}$$

But, how to obtain these models in detail?

The continuous-time models of the mechanical systems

Euler-Lagrange formalism for planar mechanical chains - computing the $\mathcal{K}$ and $\mathcal{V}$

- Lagrangian $\mathcal{L}$ requires kinetic and potential energy

$$\mathcal{L}(q, \dot{q}) = \mathcal{K} - \mathcal{V} = \frac{1}{2}\dot{q}^T D(q)\dot{q} - \mathcal{V}(q).$$

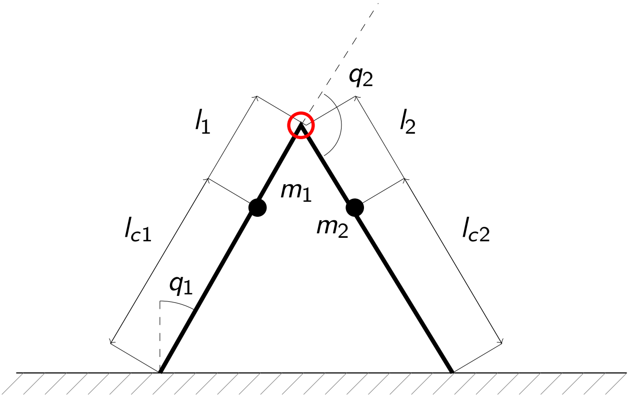
- The kinetic energy $\mathcal{K}$ of the each rigid link:

$$\mathcal{K} = \frac{1}{2}mv^{\mathrm{T}}v + \frac{1}{2}\omega^{\mathrm{T}}\mathcal{I}\omega.$$

  Here, m is the total mass of the each rigid link; v is the link center of mass (COM) velocity vector, ω is the link rotation angular velocity with respect to its COM; $\mathcal{I}$ is the symmetric 3×3 inertia tensor of the link. In 2D case, just a scalar.
- The potential energy $\mathcal{V}$ of the each rigid link: $\mathcal{V} = mgh$. Here, h is the height of the center of mass of the link.
- Choice of $q, \dot{q}$ depends on available inputs. This causes often complex $D(q)$.

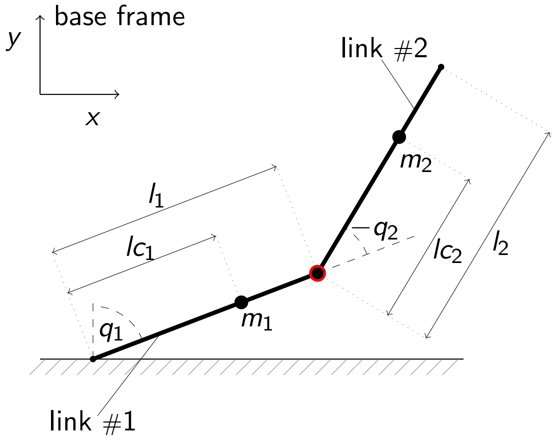
The continuous-time models of the mechanical systems

Euler-Lagrange formalism for planar mechanical chains - Acrobot (aka CGW) example.



The continuous-time models of the mechanical systems

Euler-Lagrange formalism for planar mechanical chains - Acrobot (aka CGW) example.



The continuous-time models of the mechanical systems

Euler-Lagrange formalism for planar mechanical chains - Acrobot (aka CGW) example.

$$\mathcal{K} = \frac{1}{2}\dot{q}^T D(q)\dot{q}, \;\; D(q) = \left[\begin{array}{c|r} \theta_1 + \theta_2 + 2\theta_3 \cos q_2 & \theta_2 + \theta_3 \cos q_2 \\ \theta_2 + \theta_3 \cos q_2 & \theta_2 \end{array}\right]$$

$$C(q, \dot{q}) = \begin{bmatrix} -\dot{q}_2\theta_3 \sin q_2 & -(\dot{q}_1 + \dot{q}_2)\theta_3 \sin q_2 \\ \dot{q}_1\theta_3 \sin q_2 & 0 \end{bmatrix}$$

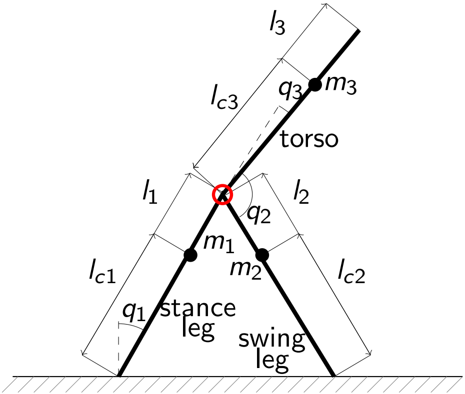
$$\mathcal{V}(q) = [\theta_4 \cos q_1 + \theta_5 \cos (q_1 + q_2)]$$

$$G(q) = \begin{bmatrix} -\theta_4 \sin q_1 - \theta_5 \sin (q_1 + q_2) \\ -\theta_5 \sin (q_1 + q_2) \end{bmatrix},$$

$$\theta_1 = m_1 l_{c1}^2 + m_2 l_1^2 + I_1, \;\; \theta_2 = m_2 l_{c2}^2 + I_2,$$
$$\theta_3 = m_2 l_1 l_{c2}, \;\; \theta_4 = g m_1 l_{c1} + g m_2 l_1, \;\; \theta_5 = g m_2 l_{c2}.$$

The continuous-time models of the mechanical systems

The three-link (aka Compass-Gait Walker with Torso)



The continuous-time models of the mechanical systems

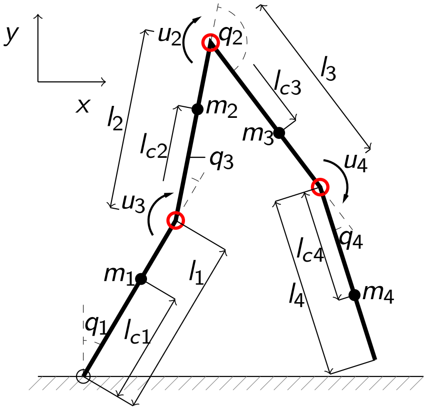
The three-link (aka Compass-Gait Walker with Torso)

Mathematical model

$$
\begin{aligned}
& D = [d_{ij}],\ \ i,j = 1,2,3,\ \ D^\top = D > 0, \quad G = [G_1, G_2, G_3]^\top, \\
& d_{11} = I_1 + I_2 + I_3 + l_1^2 m_2 + l_1^2 m_3 + l_{c1}^2 m_1 + \\
& l_{c2}^2 m_2 + l_{c3}^2 m_3 + 2 l_1 l_{c2} m_2 \cos q_2 + 2 l_1 l_{c3} m_3 \cos q_3, \\
& d_{12}(q_2) = m_2 l_{c2}^2 + l_1 m_2 \cos q_2 l_{c2} + I_2 \\
& d_{13}(q_3) = m_3 l_{c3}^2 + l_1 m_3 \cos q_3 l_{c3} + I_3,\ \ d_{23} = 0, \\
& d_{22}(q_2, q_3) = m_2 l_{c2}^2 + I_2,\ \ d_{33}(q_2, q_3) = m_3 l_{c3}^2 + I_3, \\
& G_1 = -g \left(l_1 m_2 \sin q_1 + l_1 m_3 \sin q_1 + l_{c1} m_1 \sin q_1 + \right. \\
& \left. l_{c2} m_2 \sin q_1 + q_2 + l_{c3} m_3 \sin q_1 + q_3 \right), \\
& G_2 = -g l_{c2} m_2 \sin q_1 + q_2,\ \ G_3 = -g l_{c3} m_3 \sin q_1 + q_3.
\end{aligned}
$$

The continuous-time models of the mechanical systems

The four-link model



m_1, m_4	$1\,[\mathrm{Kg}]$	m_2, m_3	$1.5\,[\mathrm{Kg}]$
l_1, l_4	$0.5\,[\mathrm{m}]$	l_{c1}, l_{c4}	$0.3\,[\mathrm{m}]$
l_2, l_3	$0.6\,[\mathrm{m}]$	l_{c2}, l_{c3}	$0.4\,[\mathrm{m}]$

The continuous-time models of the mechanical systems

The four-link model

$$D(q)=\begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ d_{31} & d_{32} & d_{33} & d_{34} \\ d_{41} & d_{42} & d_{43} & d_{44} \end{bmatrix}, C(q,\dot{q})=\begin{bmatrix} C^1 \\ C^2 \\ C^3 \\ C^4 \end{bmatrix}, G(q)=\begin{bmatrix} G_1 \\ G_2 \\ G_3 \\ G_4 \end{bmatrix},$$

$$\begin{aligned} d_{11} &= (I_1 + I_2 + I_3 + I_4 + l_1^2 m_2 + l_1^2 m_3 + l_1^2 m_4 \\ &+ l_2^2 m_3 + l_2^2 m_4 + l_3^2 m_4 + l_{c1}^2 m_1 \\ &+ l_{c2}^2 m_2 + l_{c3}^2 m_3 + l_{c4}^2 m_4 - 2 l_1 l_3 m_4 \cos(q_2 + q_3) \\ &- 2 l_1 l_{c3} m_3 \cos(q_2 + q_3) - 2 l_2 l_{c4} m_4 \cos(q_2 + q_4) \\ &+ 2 l_1 l_2 m_3 \sin(q_3) + 2 l_1 l_2 m_4 \sin(q_3) + 2 l_2 l_3 m_4 \sin(q_2) + 2 l_1 l_{c2} m_2 \sin(q_3) \\ &+ 2 l_2 l_{c3} m_3 \sin(q_2) + 2 l_3 l_{c4} m_4 \sin(q_4) - 2 l_1 l_{c4} m_4 \sin(q_2 + q_3 + q_4)) \end{aligned}$$

$$\begin{aligned} d_{12} &= (I_3 + I_4 + l_3^2 m_4 + l_{c3}^2 m_3 + l_{c4}^2 m_4 \\ &- l_1 l_3 m_4 \cos(q_2 + q_3) - l_1 l_{c3} m_3 \cos(q_2 + q_3) \\ &- l_2 l_{c4} m_4 \cos(q_2 + q_4) + l_2 l_3 m_4 \sin(q_2) + l_2 l_{c3} m_3 \sin(q_2) \\ &+ 2 l_3 l_{c4} m_4 \sin(q_4) - l_1 l_{c4} m_4 \sin(q_2 + q_3 + q_4)) \end{aligned}$$

The continuous-time models of the mechanical systems
Four-link model

$$\begin{aligned}
d_{13} = \quad & (I_2 + I_3 + I_4 + l_2^2 m_3 + l_2^2 m_4 + l_3^2 m_4 + l_{c2}^2 m_2 + l_{c3}^2 m_3 + l_{c4}^2 m_4 - \\
& l_1 l_3 m_4 \cos(q_2 + q_3) - l_1 l_{c3} m_3 \cos(q_2 + q_3) - 2 l_2 l_{c4} m_4 \cos(q_2 + q_4) + \\
& l_1 l_2 m_3 \sin(q_3) + l_1 l_2 m_4 \sin(q_3) + 2 l_2 l_3 m_4 \sin(q_2) + l_1 l_{c2} m_2 \sin(q_3) + \\
& 2 l_2 l_{c3} m_3 \sin(q_2) + 2 l_3 l_{c4} m_4 \sin(q_4) - l_1 l_{c4} m_4 \sin(q_2 + q_3 + q_4))
\end{aligned}$$

$$\begin{aligned}
d_{14} = \quad & (I_4 + l_{c4}^2 m_4 - l_2 l_{c4} m_4 \cos(q_2 + q_4) + \\
& l_3 l_{c4} m_4 \sin(q_4) - l_1 l_{c4} m_4 \sin(q_2 + q_3 + q_4))
\end{aligned}$$

$$d_{22} = \quad (m_4 l_3^2 + 2 m_4 \sin(q_4) l_3 l_{c4} + m_3 l_{c3}^2 + m_4 l_{c4}^2 + I_3 + I_4)$$

$$\begin{aligned}
d_{23} = \quad & (m_4 l_3^2 + 2 m_4 \sin(q_4) l_3 l_{c4} + l_2 m_4 \sin(q_2) l_3 + m_3 l_{c3}^2 + l_2 m_3 \sin(q_2) l_{c3} + \\
& m_4 l_{c4}^2 - l_2 m_4 \cos(q_2 + q_4) l_{c4} + I_3 + I_4)
\end{aligned}$$

$$d_{24} = \quad (m_4 l_{c4}^2 + l_3 m_4 \sin(q_4) l_{c4} + I_4)$$

$$\begin{aligned}
d_{33} = \quad & (I_2 + I_3 + I_4 + l_2^2 m_3 + l_2^2 m_4 + l_3^2 m_4 + \\
& l_{c2}^2 m_2 + l_{c3}^2 m_3 + l_{c4}^2 m_4 - 2 l_2 l_{c4} m_4 \cos(q_2 + q_4) + 2 l_2 l_3 m_4 \sin(q_2) + \\
& 2 l_2 l_{c3} m_3 \sin(q_2) + 2 l_3 l_{c4} m_4 \sin(q_4))
\end{aligned}$$

$$d_{34} = \quad (I_4 + l_{c4}^2 m_4 - l_2 l_{c4} m_4 \cos(q_2 + q_4) + l_3 l_{c4} m_4 \sin(q_4))$$

$$d_{44} = \quad (m_4 l_{c4}^2 + I_4)$$

The continuous-time models of the mechanical systems

Four-link model

$$
\begin{aligned}
G_1 = \quad & -gl_{c4}m_4\sin(q_1+q_2+q_3+q_4) - gl_2m_3\sin(q_1+q_3) - gl_2m_4\sin(q_1+q_3) \\
& -gl_{c2}m_2\sin(q_1+q_3) - gl_1m_2\sin(q_1) - gl_1m_3\sin(q_1) - gl_1m_4\sin(q_1) - \\
& gl_{c1}m_1\sin(q_1) - gl_3m_4\sin(q_1+q_2+q_3) - gl_{c3}m_3\sin(q_1+q_2+q_3) \\
G_2 = \quad & -gl_{c4}m_4\sin(q_1+q_2+q_3+q_4) - gl_3m_4\sin(q_1+q_2+q_3) \\
& -gl_{c3}m_3\sin(q_1+q_2+q_3) \\
G_3 = \quad & -gl_{c4}m_4\sin(q_1+q_2+q_3+q_4) - gl_2m_3\sin(q_1+q_3) - \\
& gl_2m_4\sin(q_1+q_3) - gl_{c2}m_2\sin(q_1+q_3) \\
& -gl_3m_4\sin(q_1+q_2+q_3) - gl_{c3}m_3\sin(q_1+q_2+q_3) \\
G_4 = \quad & -gl_{c4}m_4\sin(q_1+q_2+q_3+q_4).
\end{aligned}
$$

The hybrid models of the mechanical systems

Swing and impact phases

- Contact mechanical systems are modeled using both continuous-time and discrete-time dynamics.
- Hybrid systems combine both dynamics:
  - **continuous-time** dynamics

$$\dot{x} = F(x, u), \quad x \in \mathcal{C}$$

  - **discrete-time** dynamics.

$$x^{+} = \Gamma(x^{-}, u), \quad x \in \mathcal{D}$$

  Usually, $\mathcal{D}$ is some lower dimensional submanifold of $\mathcal{C}$.

- For walking $\Gamma(x^{-}, u) \equiv \Gamma(x^{-})$, as only impulsive forces acting.

The hybrid models of the mechanical systems

Swing and impact phases

- Actually: $\dot{q}^+ = \Phi(q^-)\dot{q}^-$ and q^+ undergoes some simple **relabeling map** due to switching the legs. Φ is called as the **impact matrix**.
- Switching of legs is to keep the same continuous time model for both legs being the swing one. Alternative would be hybrid systems with two continuous-time models.
- Both leg are usually assumed to have the same properties.

Discrete-time dynamics modeling

Impact matrix modeling

- When the swing leg of the Acrobot hits the ground, the impact occurs.
- Impact causes instantaneous jump in angular velocities $\dot{q}$ while angular positions q remain continuous in time.
- The impact is modeled as a contact between two rigid bodies:
  - double support phase is instantaneous,
  - overall energy and momentum is preserved,
  - no swing leg rebound,
  - no swing leg slipping.
- The impact modeling is based on the continuous-time models shortly "just before the impact" and "just after the impact".

Discrete-time dynamics modeling

Impact matrix modeling

- The extended continuous-time model is needed that unifies both situations. It has more DOF generalized coordinates denoted q_e, its matrix of inertia denoted $D_e(q_e)$.
- The impact matrix computation is based on the equations:

$$D_e\left[\dot{q}_e^+ - \dot{q}_e^-\right] = F_{ext}, \quad E_2(q_e^-)\dot{q}_e^+ = 0,$$

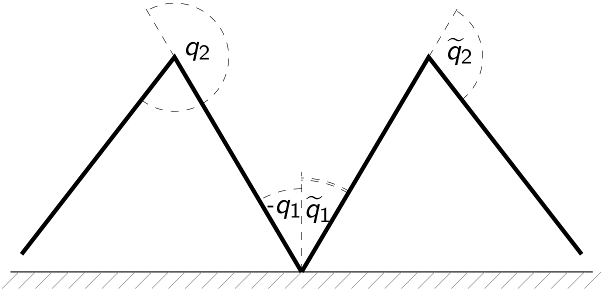
where $E_2(q_e^-) = \frac{\partial \Upsilon(q_e)}{\partial q_e}(q_e^-)$, Υ represents swing leg's end point Carthesian coordinates, q_e^- corresponds to the double support configuration. Vector F_{ext} is the assumed cumulative effect of the impulsive forces during the infinitesimally small time interval.

- F_{ext} is unknown, but can it be eliminated.
  E.g., for Acrobot there are 10 scalar variables: $\dot{q}_e^-, \dot{q}_e^+, F_{ext}$ and 6 equations. So, one can obtain 4 linear equations relating $\dot{q}_e^-$ and $\dot{q}_e^+$, *i.e.*, consequently, two linear equations for $\dot{q}^-, \dot{q}^+$ in the form $\dot{q}^+ = \Phi(q^-)\dot{q}^-$.

Discrete-time dynamics modeling

Relabeling map

- At impact, the swing leg, respectively stance leg, becomes the new stance leg, respectively the new swing leg.
- Example: the Acrobot's relabeling of q_1 and q_2 coordinates.



- This picture also helps to undestand the essence and the need of that previously mentioned extended model.

Virtual holonomic constraints

Definition and regularity

Virtual holonomic constraints (VHC) of $q \in R^n$ are equalities

$$\varphi_1(q) = \varphi_2(q) = \ldots = \varphi_l(q) = 0.$$

Smooth functions $\varphi_1, \ldots, \varphi_l$ satisfy $\mathrm{rank}\{\mathrm{d}\varphi_1(q), \ldots, \mathrm{d}\varphi_l(q)\} = l$ $\forall q \in \{q \in R^n | \varphi_1(q) = \cdots = \varphi_l(q) = 0\}$.

VHC are called global if $\varphi_1(q), \ldots, \varphi_l(q)$ can be completed to a global diffeomorphism of R^n.

VHC Purpose: Design problem decomposed: **a)** VHC enforcement; **b)** design for the less DOF constrained dynamics

Regular VHC: can be enforced by feedback control

Further peculiarity: Hybrid invariance needed

Realization of the virtual holonomic constraints

Regular VHC and input-output exact feedback linearization

$$\dot{x} = f(x) + u_1 g^1(x) + \ldots + u_m g^m(x),\ y = [h^1(x), \ldots, h^p(x)],\ m \geq p,$$

$$x \in R^n, y \in \mathbb{R}^p, u \in R^m,\ f(x_0) = 0, \mathrm{rank}[g^1| \cdots |g^m](x_0) = m, h(x_0 = 0.$$

Lie derivative:

$$L_f h := f_1 \frac{\partial h}{\partial x_1} + \ldots + f_n \frac{\partial h}{\partial x_n},\quad L_f^0 h := h,\ \ L_f^{k+1} h := L_f L_f^k h, \forall k = 1, 2 \ldots$$

Vector relative degree $(r_1, \ldots, r_p)$:

$$L_{g^i} L_f^k h_j \equiv 0,\quad \forall k = 0, \ldots, r_j - 2,\ i = 1, \ldots, m,\ j = 1, \ldots, p \text{ and}$$

$$\mathrm{rank}\mathcal{D}(x_0) = p,\quad \mathcal{D}(x) := \begin{bmatrix} L_{g^1} L_f^{r_1 - 1} h_1 & \ldots & L_{g^m} L_f^{r_1 - 1} h_1 \\ \vdots & \vdots & \\ L_{g^1} L_f^{r_p - 1} h_p & \ldots & L_{g^m} L_f^{r_p - 1} h_p \end{bmatrix} (x)$$

$\mathcal{D}$ is called as the **decoupling matrix**.

Realization of the virtual holonomic constraints

Regular VHC and input-output exact feedback linearization

It holds (here $(\cdot)^{(r)}$ stands for the r-th order time derivative):

$$\begin{bmatrix} y_1^{(r_1)}(t) \\ \vdots \\ y_p^{(r_p)}(t) \end{bmatrix} = \mathcal{D}(x) \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix} + \begin{bmatrix} L_f^{r_1} h_1 \\ \vdots \\ L_f^{r_p} h_p \end{bmatrix}.$$

Moreover, there are $r_1 + r_2 + \ldots + r_n$ independent functions:

$$y_1 = h_1(x),\ y_1^{(1)} = L_f h_1(x), \ldots, y_1^{(r_1-1)} = L_f^{r_1-1} h_1(x),\ \ldots$$

$$y_p = h_p(x),\ y_2^{(1)} = L_f h_p(x), \ldots, y_p^{(r_p-1)} = L_f^{r_p-1} h_p(x).$$

These function can be used as a part of new coordinates, giving $(r_1 + r_2 + \ldots + r_n)$-dimensional linear subsystem, consisting from l independent integrators chains having lengths $r_1, \ldots, r_p$.

Realization of the virtual holonomic constraints

Regular VHC and input-output exact feedback linearization

Note that:

$$\mathcal{D}(x) = \frac{\partial}{\partial u} \begin{bmatrix} \ddot{y}_1(t) \\ \vdots \\ \ddot{y}_l(t) \end{bmatrix}$$

In other words, regular VHC are such that the mechanical system with $n-k$ inputs $u_{n-k+1}, \dots u_n$ and l outputs

$$y_1 = \varphi_1(q),\ y_2 = \varphi_2(q),\ \dots,\ y_l = \varphi_l(q)$$

has the vector relative degree $(2, \dots, 2)$.
Therefore, the assumption $l \leq n-k$ is needed. One can enforce at most as much constraints as it is the number of inputs.

Realization of the virtual holonomic constraints

Regular VHCs and input-output exact feedback linearization

Realization can be done *e.g.* by

$$\ddot{y}_1 = -k_1^1 y_1 - k_1^2 \dot{y}_1, \dots, \ddot{y}_l = -k_l^1 y_l - k_l^2 \dot{y}_l,$$

where all k's are positive reals. Recall, that
$y_1 = \varphi_1(q),\ y_2 = \varphi_2(q),\ \dots,\ y_l = \varphi_l(q)$ and therefore also

$$\dot{y}_1 = \frac{\partial \varphi_1}{\partial q}\dot{q}, \dots, \dot{y}_l = \frac{\partial \varphi_l}{\partial q}\dot{q}.$$

$$x := (q^\top, \dot{q}^\top)^\top, f(x) = (x_1, \dots, x_n, -D^{-1}(C\dot{q} - G)^\top)^\top,\ G = D^{-1},$$

$$\begin{bmatrix} \ddot{y}_1 \\ \vdots \\ \ddot{y}_l \end{bmatrix} = \mathcal{D}(q, \dot{q})u + \begin{bmatrix} L_f^2 \varphi_1 \\ \vdots \\ L_f^2 \varphi_l \end{bmatrix} \implies u = [\mathcal{D}(q, \dot{q})]^{-1} \begin{bmatrix} \ddot{y}_1 - L_f^2 \varphi_1 \\ \vdots \\ \ddot{y}_l - L_f^2 \varphi_l \end{bmatrix}$$

Sensors, estimation and identification

Sensors

All previous approaches to be implemented require state estimation, or direct measurements of all states.

- Angular positions measurement at rotary joints efficient and almost noisy free.
- Relative measurements: only increments of angle relative to their initial value measured, *e.g.* IRC (Incremental Rotary enCoder) - usually optical, very good precision and low noise.
- Absolute measurements: absolute angle, *e.g.* potentiometer (simple, cheap, but low precision), magnetic position sensor (Hall effect sensor, much better, but not as IRC), ...
- Angle not related to any rotary joint, *e.g.* the angle at the pivot point (or any other absolute orientation angle), measured indirectly only, if the robot is autonomous: inertial sensors, digital gyroscopes, laser distance sensors...

Sensors, estimation and identification

State estimation

- Velocities either measured directly by gyroscopes, or estimated.
- For noisy-free angular positions measurements (*e.g.* IRC), numerical time derivative applicable, or filtered numerical time derivatives, fast evaluation circuit needed.
- For the absolute orientation angle estimation more complicated.
- In 2D-walking platforms often noisy-free measurement of the absolute orientation angle implemented in supporting platform.
- Kalman filtering and other methods from control courses used as well, with some adaptation (Extended Kalman Filter, Hybrid Kalman Filter, ...).

Sensors, estimation and identification

Identification

- Mechanical parameters can be well-measured in advance (weights, lengths, moments of inertia).
- These measurements may serve for further tuning as the parameters initial estimates.
- Noise in drives effects need to be attenuated.
- If all angles are well measured, angular velocities and angular accelerations well computed, estimation of $\theta_1, \theta_2, \ldots$ becomes a standard linear problem, *e.g.* least squares and maximal likehood method applicable.
- Again, there is a problem with absolute orientation angle. But it can be handled easier, than in state estimation problem, as for identification off-line experiments possible, using some frames and platforms with extra measurements,...

Sensors, estimation and identification

Identification - Acrobot (aka CGW) example

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \begin{bmatrix} 0 \\ u_2 \end{bmatrix}$$

$$C(q,\dot{q}) = \begin{bmatrix} -\dot{q}_2\theta_3 \sin q_2 & -(\dot{q}_1 + \dot{q}_2)\theta_3 \sin q_2 \\ \dot{q}_1\theta_3 \sin q_2 & 0 \end{bmatrix}$$

$$G(q) = \begin{bmatrix} -\theta_4 \sin q_1 - \theta_5 \sin(q_1 + q_2) \\ -\theta_5 \sin(q_1 + q_2) \end{bmatrix},$$

$$\theta_1 = m_1 l_{c1}^2 + m_2 l_1^2 + I_1,\ \theta_2 = m_2 l_{c2}^2 + I_2,$$
$$\theta_3 = m_2 l_1 l_{c2},\ \theta_4 = g m_1 l_{c1} + g m_2 l_1,\ \theta_5 = g m_2 l_{c2}.$$

Related skipped problems

Running robots

Jumping robots

And many others ...

Note, that running and jumping may be in a certain sense easier than walking (compare to track and field athletes experience!)

Mathematical modeling explanation: no need for the complex continuous-time walking-like dynamics, with no contact with ground, robot is governed just by laws of passive free movement under the gravity and air resistance influence.

Literature

- Ch. Chevallereau, G. Abba, Y. Aoustin, F. Plestan, E. R. Westervelt, C. Canudas-De-Wit, and J. W. Grizzle. Rabbit: a testbed for advanced control theory. IEEE Control Systems, 23 (5), 57–79, 2003.
- E. R. Westervelt and J. W. Grizzle and Ch. Chevallereau and J. H. Choi and B. Morris: *Feedback Control of Dynamic Bipedal Robot Locomotion*, CRC Press, 2007.
- Ch. Chevallereau, G. Bessonnet, G. Abba, and Y. Aoustin. Bipedal Robots: Modeling, Design and Walking Synthesis. Wiley-ISTE, 2009.
- J.W. Grizzle, C. Chevalereau, R.W. Sinnet, and A.D. Ames. Models, feedback control and open problems of 3D bipedal robotic walking. Automatica, 50 (8), 1955–1988, 2014.
- Surbhi Gupta and Amod Kumar. A brief review of dynamics and control of underactuated biped robots. Advanced Robotics, 31 (12), 607–623, 2017.
- Lecture of J.W. Grizzle at MIT, October 2019 https://www.youtube.com/watch?v=rVPVZn2aNWI