

机器人导论



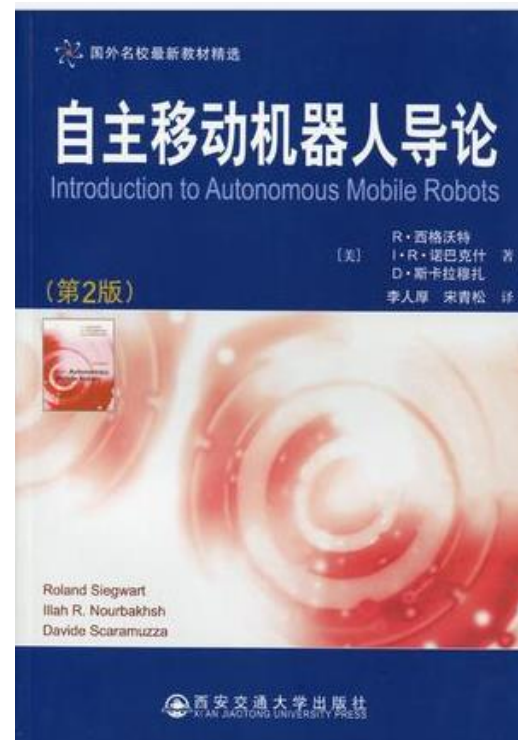
机器人导论：智能科学与技术、自动化专业选修课

教学内容：机器人的基本结构组成、工作原理、控制、感知、路径规划

教学目标：使学生通过本课程的学习，了解当前国内外机器人控制的研究、应用状况以及发展趋势，初步具备机器人系统的设计能力，提高学生分析问题和解决问题的能力

参考教材：Introduction to Autonomous Mobile Robot, Roland Siegwart, ETH Zurich, 2011

《自主移动机器人导论》，李人厚译，西安交通大学出版社，2013年5月



机器人的控制



AlphaDog



Atlas



Hubo



CHIMP



打磨机器人



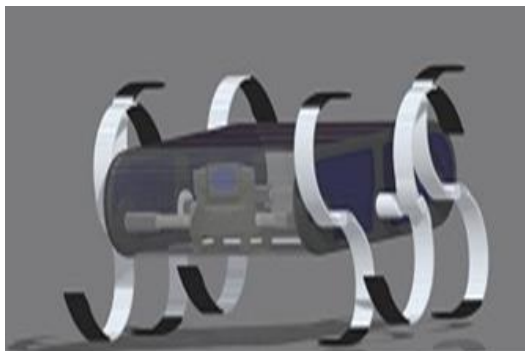
Roomaba



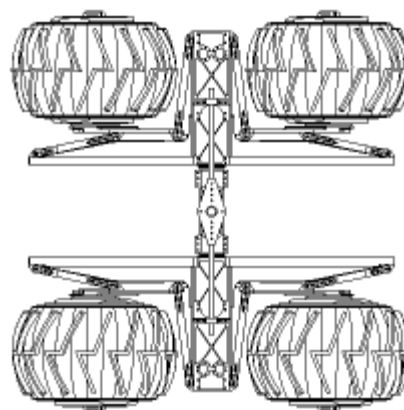
furo



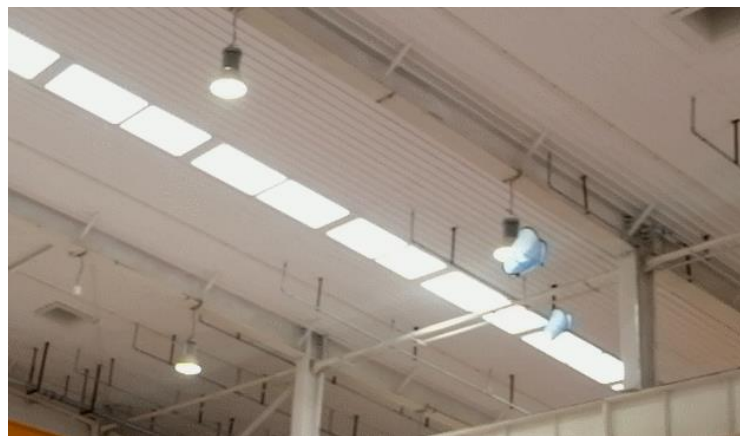
ROVIO



机器人的控制



<https://www.festo.com/group/en/cms/12747.htm>

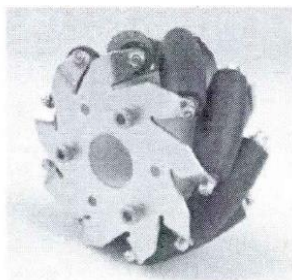


全向移动机器人



全向移动机器人灵活、自由度高

- 三轮全向移动机器人
- 四轮全向移动机器人



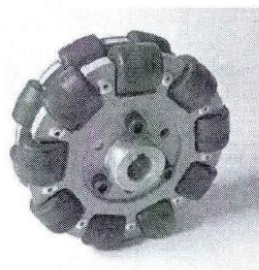
(a) Mecanum 轮



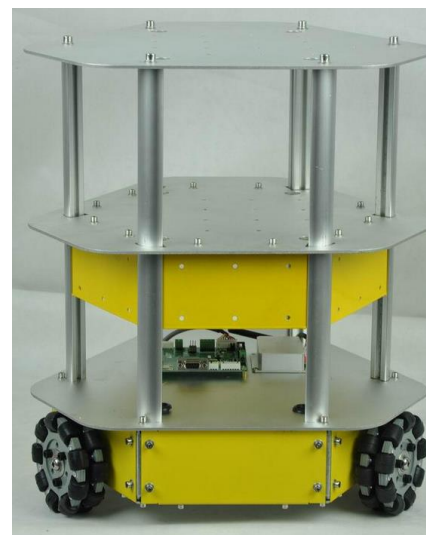
(b) 球形轮



(c) 单层 Grabowiecki 轮



(d) 双层 Grabowiecki 轮



全向移动机器人



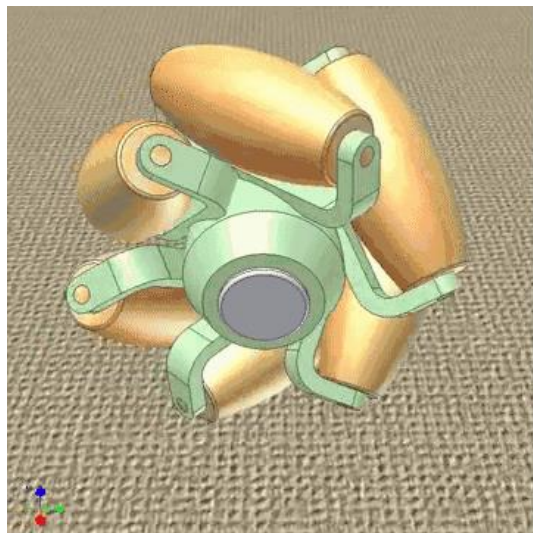
KUKA omniMove

KUKA omniMove 是我们的重载型运输平台。可以非常方便地将其移至正确位置——在最狭窄的空间内并以毫米级精度。此外，KUKA omniMove 的灵活性高，可以方便地用遥控器控制。



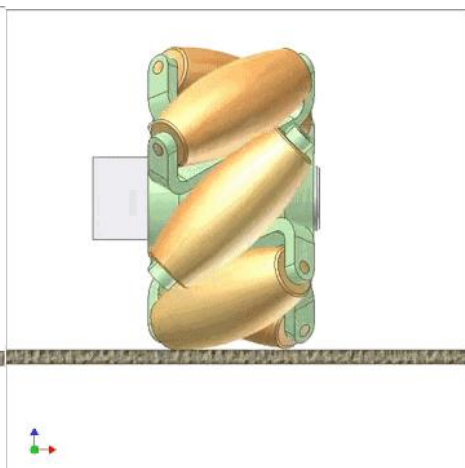
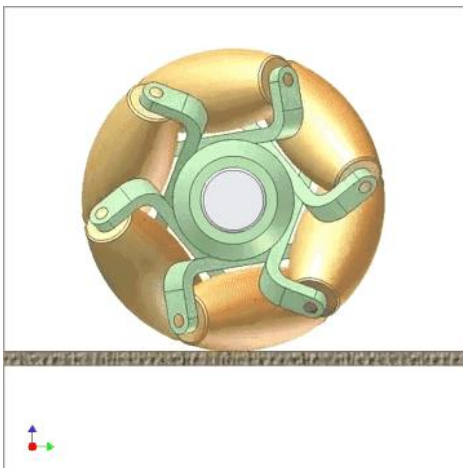
全向车轮技术可以使车轮沿任意方向移动。一个车轮由两个轮辋和九个以 45 度角安装的自由滚轮。各个车轮的运动互不相干，使车辆不仅可以前进和侧移，而且还可以沿对角线运动和做圆周运动。每个车轮由电机单独驱动（四驱）。单个滚轮可以在车辆上直接更换，无需拆下整个车轮，几分钟即可完成。

全向移动机器人：麦克纳姆轮

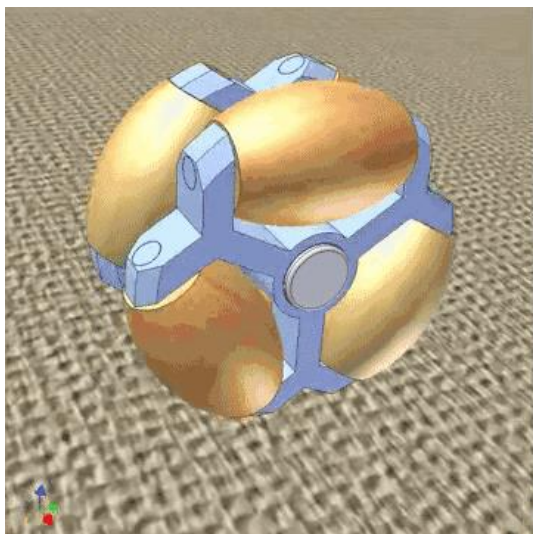


麦克纳姆轮

1973年瑞士发明家Bengt Ilon在名为Mecanum的公司，发明可以任意自由方向移动的平台。他的构想是在车轮外环中固定与轴心成 45° 的自由滚子，自由滚子的排列有两种不同，一个是外向 45° 一个是内向 45° 。车轮旋转时成 45° 排列的自由滚子与地面接触，地面会给予车轮与转轴夹 45° 的摩擦力，此摩擦力可分为X分量与Y分量，藉由车轮的正反转或停止，改变XY分量力的方向，可让平台做各种方式的移动。



全向移动机器人：全向轮

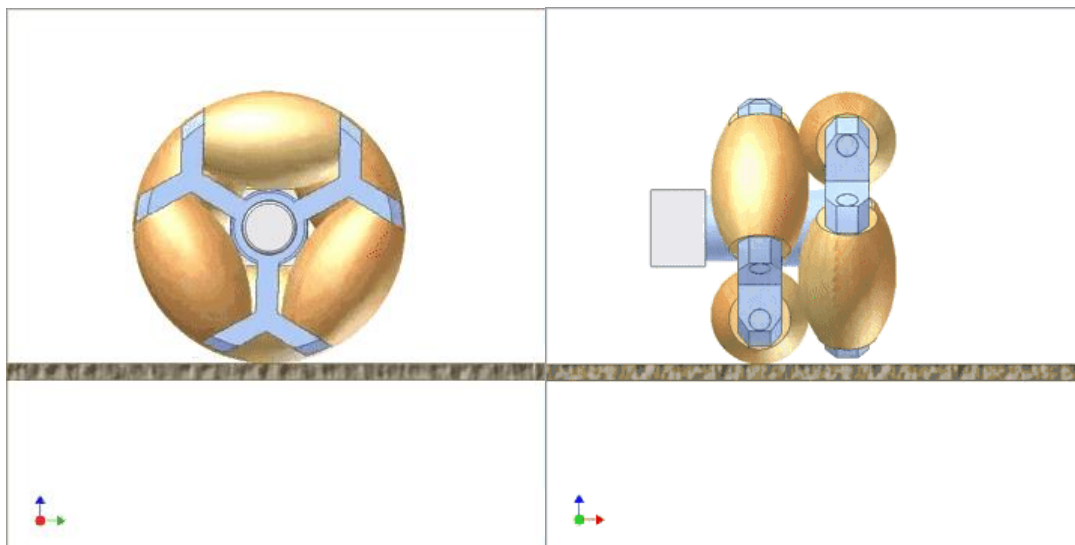


全向轮

全向轮又被称为Omni轮 (omnidirectional wheel)

该轮于1919年由J. Grabowiecki首次获得专利

美国专利1305535, J. Grabowiecki, Vehicle wheel,
1919-06-03发布



全向移动机器人：全向轮



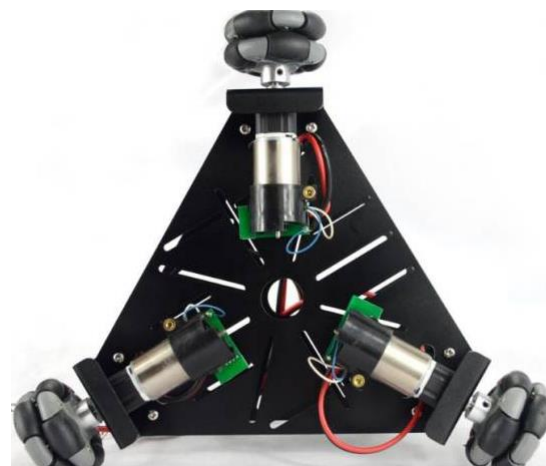
全向轮 (omnidirectional wheel)

美国专利3789947, Josef F. Blumrich, “Omnidirectional wheel”, 1974年2月5日授权

由一个轮盘和固定在轮盘外周的滚子构成。轮盘轴心同滚子轴心垂直，轮盘绕轴心由电机驱动转动，滚子依次与地面接触，并可绕自身轴心自由转动。全向轮包括轮毂和从动轮，该轮毂的外圆周处均匀开设有3个或3个以上的轮毂齿，每两个轮毂齿之间装设有一从动轮



十字形

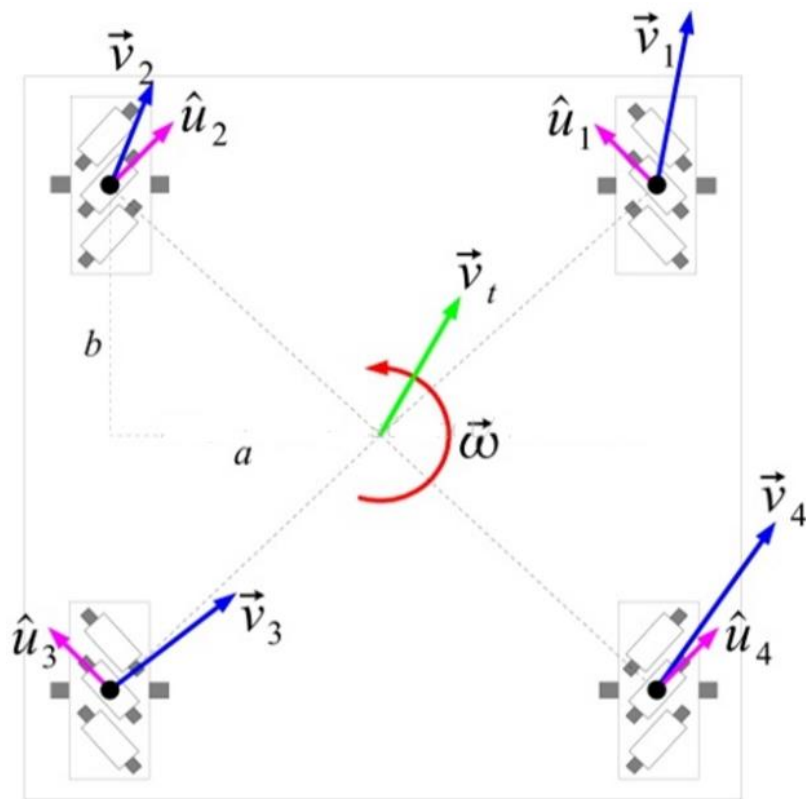
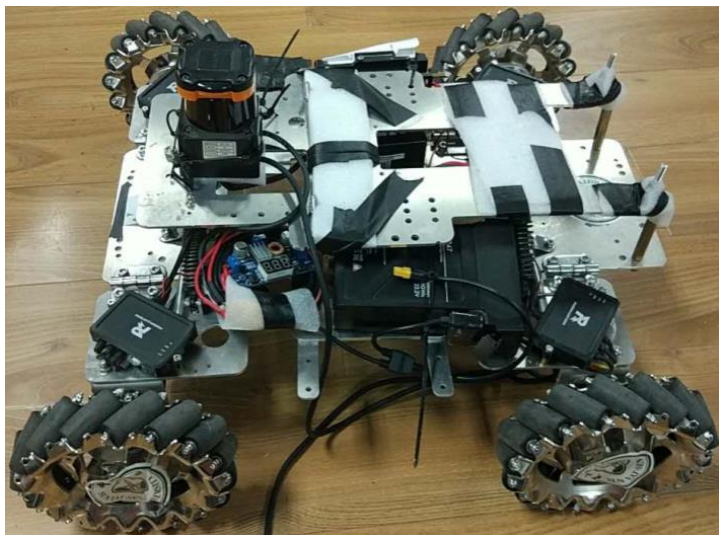
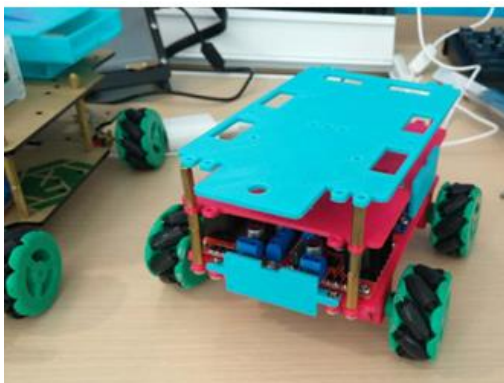


三角形

全向移动机器人：运动学模型

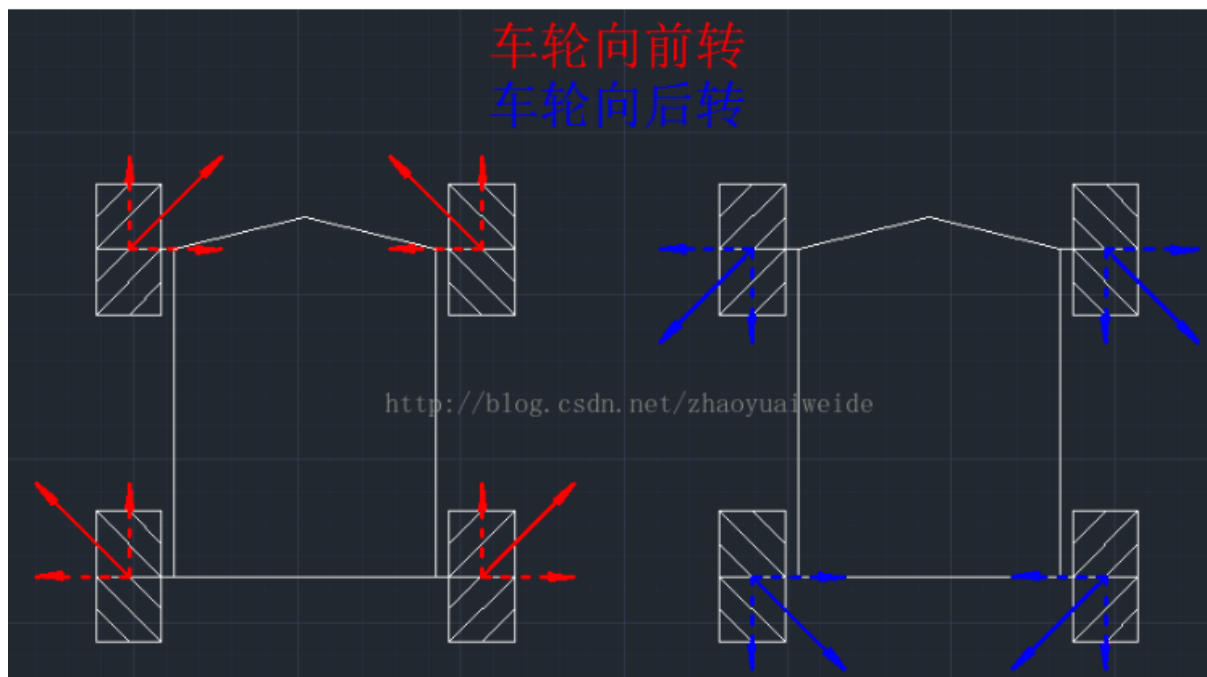


麦克纳姆轮



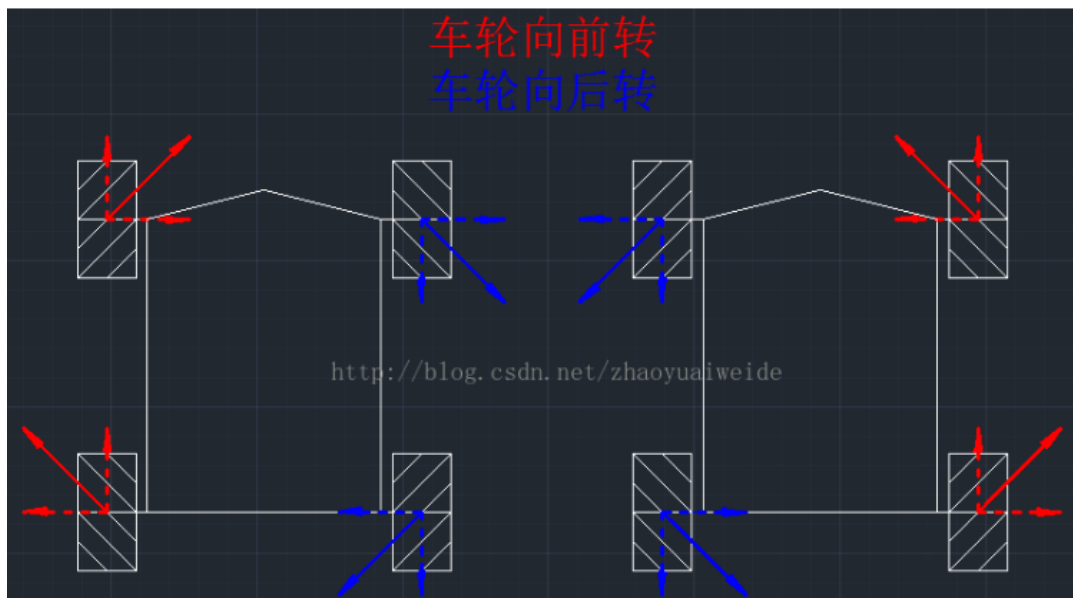
通常用长方形安装方式，四个轮子的着地点形成一个矩形，轮子转动可以产生Z轴转动力矩

全向移动机器人：运动学模型

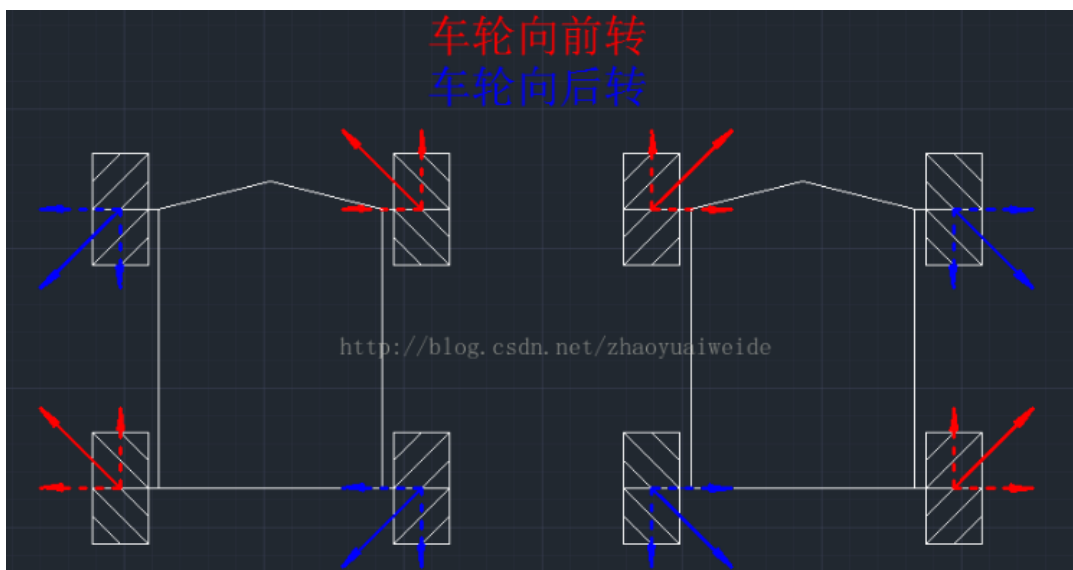


- 红色实线箭头是车轮向前转产生的摩擦力
- 蓝色箭头是车轮向后转产生的摩擦力
- 虚线是分力
- 左边的车向前跑，右边的车向后跑

全向移动机器人：运动学模型

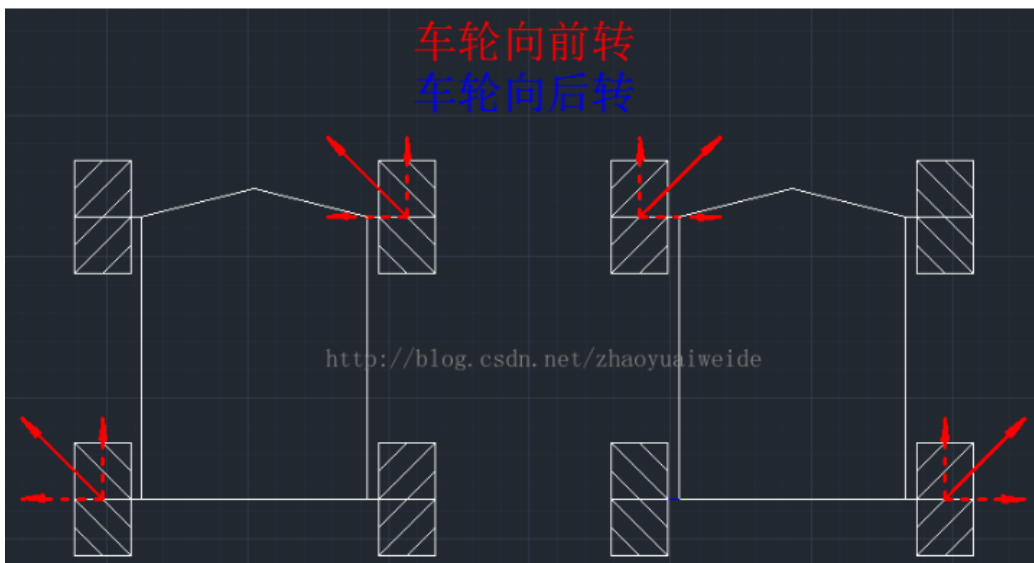


- 左边的车顺时针旋转
- 右边的车逆时针旋转

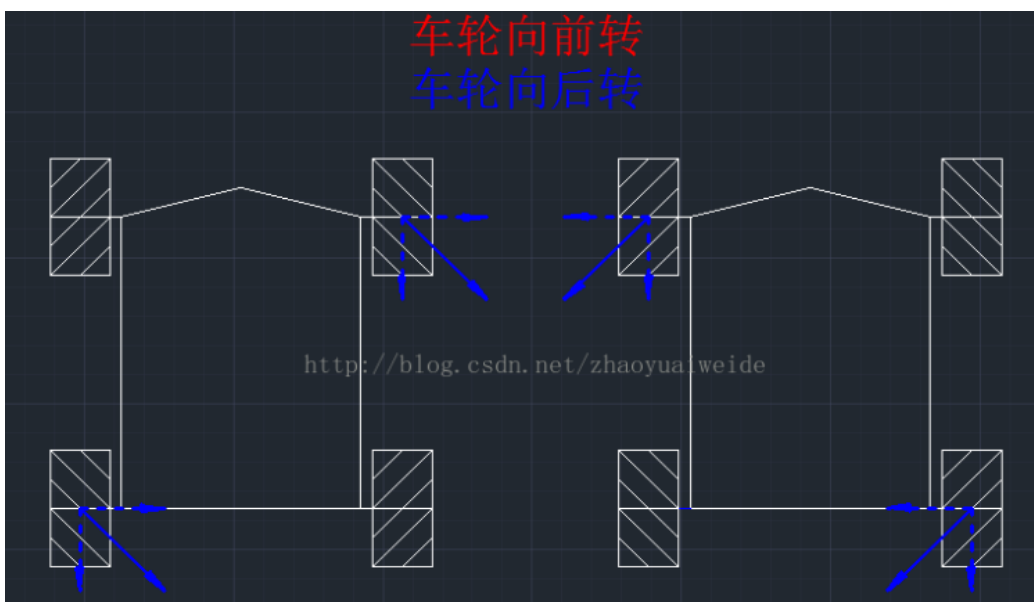


- 左边的车向左平移
- 右边的车向右平移

全向移动机器人：运动学模型



- 左边的车左上斜走
- 右边的车右上斜走



- 左边的车右下斜走
- 右边的车左下斜走

全向移动机器人：运动学模型



- ✦ 正运动学模型（forward kinematic model）：通过四个轮子的速度计算底盘的运动状态；
- ✦ 逆运动学模型（inverse kinematic model）：根据底盘的运动状态解算四个轮子的速度
- ✦ 底盘的运动用三个独立变量来描述：X轴平动、Y轴平动、Z轴自转；而四个麦轮的速度由四个独立的电机提供；因此，四个麦轮的合理速度存在某种约束关系，逆运动学可以得到唯一解，而正运动学中不符合这个约束关系的方程将无解
- ✦ 底盘运动的分解

刚体在平面内的运动可分解为三个独立分量：X轴平动、Y轴平动、Z轴自转。

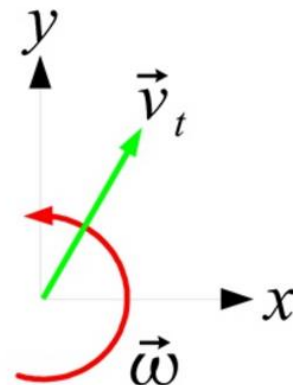
底盘的运动可分解为三个量 $[v_{tx} \ v_{ty} \ \omega]$ ：

v_{tx} ：X 轴运动的速度，即左右方向，定义向右为正；

v_{ty} ：Y 轴运动的速度，即前后方向，定义向前为正；

ω ：Z 轴自转的角速度，定义逆时针为正。

以上三个量视为四个轮子的几何中心的速度。

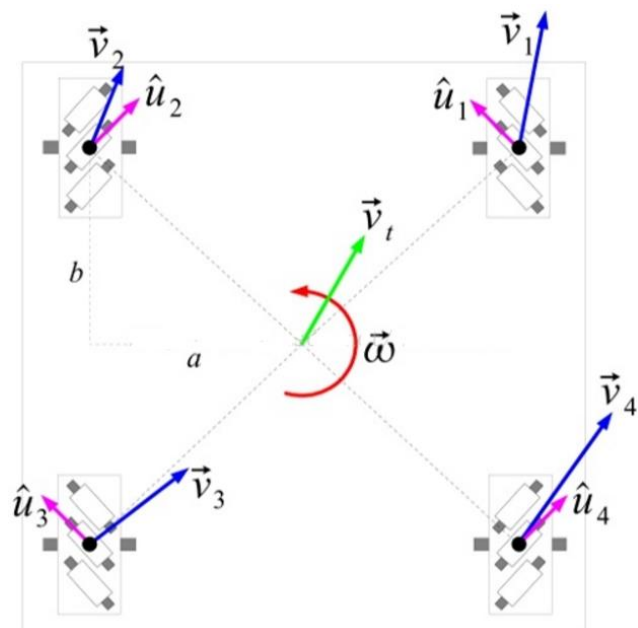


全向移动机器人：运动学模型



全向移动底盘的运动可以线性分解为三个分量，计算出麦轮底盘在“沿X轴平移”、“沿Y轴平移”、“绕Z轴自旋”时四个轮子的速度，则可计算出这三种运动所合成的“平动+旋转”时，四个车轮所需线速度 $v_{\omega 1}, v_{\omega 2}, v_{\omega 3}, v_{\omega 4}$

$$\begin{aligned} \text{底盘沿X轴平移: } & \begin{cases} v_{\omega 1} = -v_{tx} \\ v_{\omega 2} = +v_{tx} \\ v_{\omega 3} = -v_{tx} \\ v_{\omega 4} = +v_{tx} \end{cases} \\ \text{底盘沿Y轴平移: } & \begin{cases} v_{\omega 1} = v_{ty} \\ v_{\omega 2} = v_{ty} \\ v_{\omega 3} = v_{ty} \\ v_{\omega 4} = v_{ty} \end{cases} \\ \text{底盘沿Z轴自旋: } & \begin{cases} v_{\omega 1} = +\omega(a+b) \\ v_{\omega 2} = -\omega(a+b) \\ v_{\omega 3} = -\omega(a+b) \\ v_{\omega 4} = +\omega(a+b) \end{cases} \end{aligned}$$



全向移动机器人：运动学模型



将以上三种运动对应的方程组相加，则可根据底盘运动状态解算出四个轮子的速度 $v_{\omega 1}, v_{\omega 2}, v_{\omega 3}, v_{\omega 4}$

$$\begin{cases} v_{\omega 1} = v_{ty} - v_{tx} + \omega(a+b) \\ v_{\omega 2} = v_{ty} + v_{tx} - \omega(a+b) \\ v_{\omega 3} = v_{ty} - v_{tx} - \omega(a+b) \\ v_{\omega 4} = v_{ty} + v_{tx} + \omega(a+b) \end{cases} \longrightarrow \begin{bmatrix} v_{\omega 1} \\ v_{\omega 2} \\ v_{\omega 3} \\ v_{\omega 4} \end{bmatrix} = J \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} \quad (1)$$

其中, $[v_{tx} \ v_{ty} \ \omega]$ 为机器人在世界坐标系中的运动分量

v_{tx} : X 轴运动的速度, 即左右方向, 定义向右为正;

v_{ty} : Y 轴运动的速度, 即前后方向, 定义向前为正;

ω : Z 轴自转的角速度, 定义逆时针为正。

全向移动机器人：运动学模型

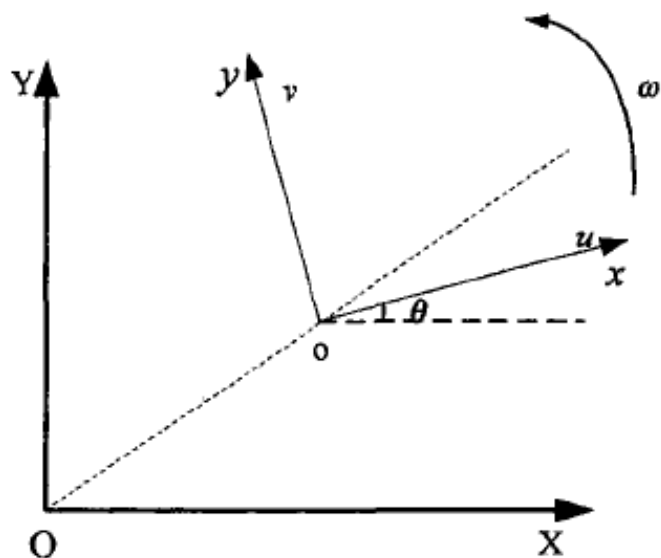


图1 机器人的坐标变换关系

设在任意一个时刻, 全向机器人在世界坐标系和本体坐标系质心的位姿分别为 $(X, Y, \theta)^T$ 和 $(x, y, \phi)^T$, 则它在世界坐标系中的速度向量为 $(\dot{X}, \dot{Y}, \dot{\theta})^T$, 取其在本体坐标系速度向量为 $(u, v, \omega)^T = (\dot{x}, \dot{y}, \dot{\phi})^T$, 则由图1的几何关系可得机器人的运动学关系为:

$$\begin{cases} \dot{X} = u \cos \theta - v \sin \theta \\ \dot{Y} = u \sin \theta + v \cos \theta \\ \dot{\theta} = \omega \end{cases} \quad (2)$$

XOY: 机器人的世界坐标系

xoy: 机器人的本体坐标系

式中, θ 代表机器人的运动方向, 为机器人本体坐标系x轴正方向与世界坐标系X轴正方向之间的夹角

全向移动机器人：运动学位姿误差模型



令全向机器人在其世界坐标系的位姿量为 $\mathbf{q} = (X, Y, \theta)^T$ ，
在本地坐标系中的速度量为 $\boldsymbol{\mu} = [u, v, \omega]^T$ ，则式(1)可进一步
写为矩阵的形式：

$$\dot{\mathbf{q}} = R(\theta)\boldsymbol{\mu} \quad (3)$$

其中旋转矩阵 $R(\theta)$ 为

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{由 (1) } \begin{bmatrix} v_{\omega 1} \\ v_{\omega 2} \\ v_{\omega 3} \\ v_{\omega 4} \end{bmatrix} = J \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} \text{ 及 (3) 可得 } \begin{bmatrix} v_{\omega 1} \\ v_{\omega 2} \\ v_{\omega 3} \\ v_{\omega 4} \end{bmatrix} = J \cdot R(\theta) \cdot \boldsymbol{\mu} \quad (4)$$

全向移动机器人：运动学位姿误差模型



当确定期望速度以后, 可以根据式 (4) 求得机器人四个轮子的转速, 进而通过对底层电机的转速控制实现对整个机器人的运动控制。传统的控制方案是采取每个电机独立的控制方式, 如图2所示, 这种控制方式相对于机器人速度而言属于开环控制。因此即使单个驱动电机的控制精度达到最优, 每个电机能够很好地跟踪其期望转速, 但由于场地材料不同, 四个轮子装配上存在误差, 轮子与场地之间摩擦力不一致, 四个驱动电机存在机械差异等原因, 机器人的速度跟踪也难以取得令人满意的效果。

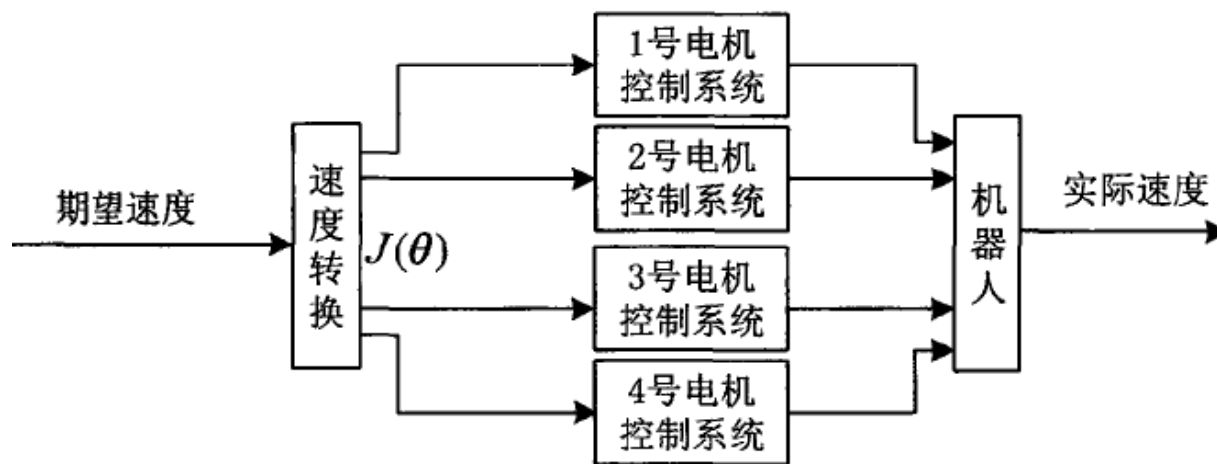


图2 传统的速度跟踪系统

全向移动机器人：运动学位姿误差模型



因此, 需设计合适的机器人速度误差校正控制器, 将其与电机控制环一起构成新的闭环控制, 如图3所示。速度误差校正控制器可为机器人的四个电机提供附加的校正量, 提高机器人对速度跟踪的控制精度。此外, 机器人的运动学方程 (3) 是在假设机器人处在理想环境下得到的, 即要求机器人的各个参数足够精确。但由于装配误差等原因, 运动学方程中的参数及转换矩阵的数值存在误差, 因此需要设计具有一定自适应能力的速度误差校正控制器以克服模型参数不精确带来的影响。

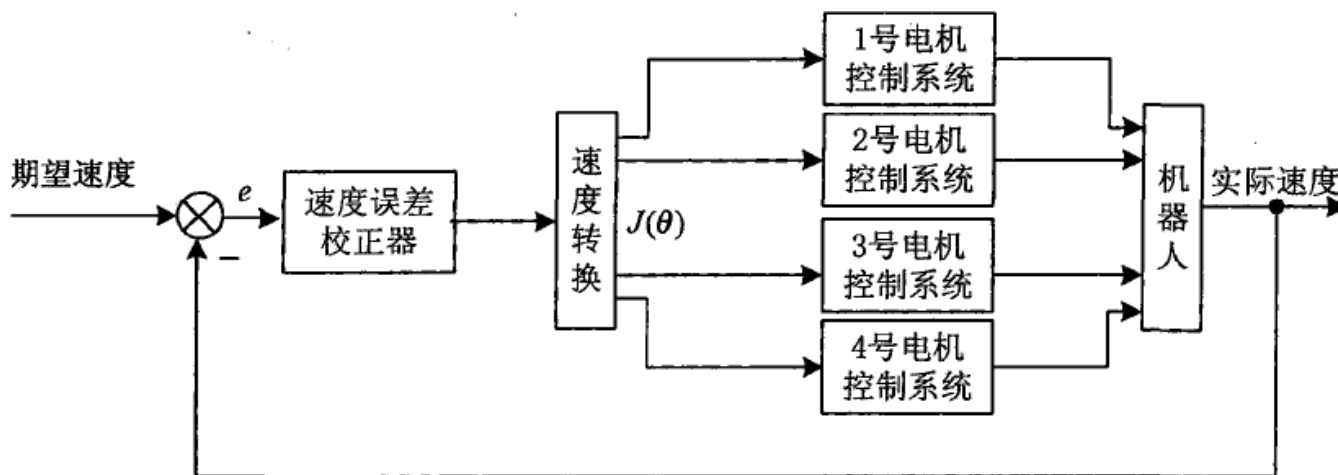
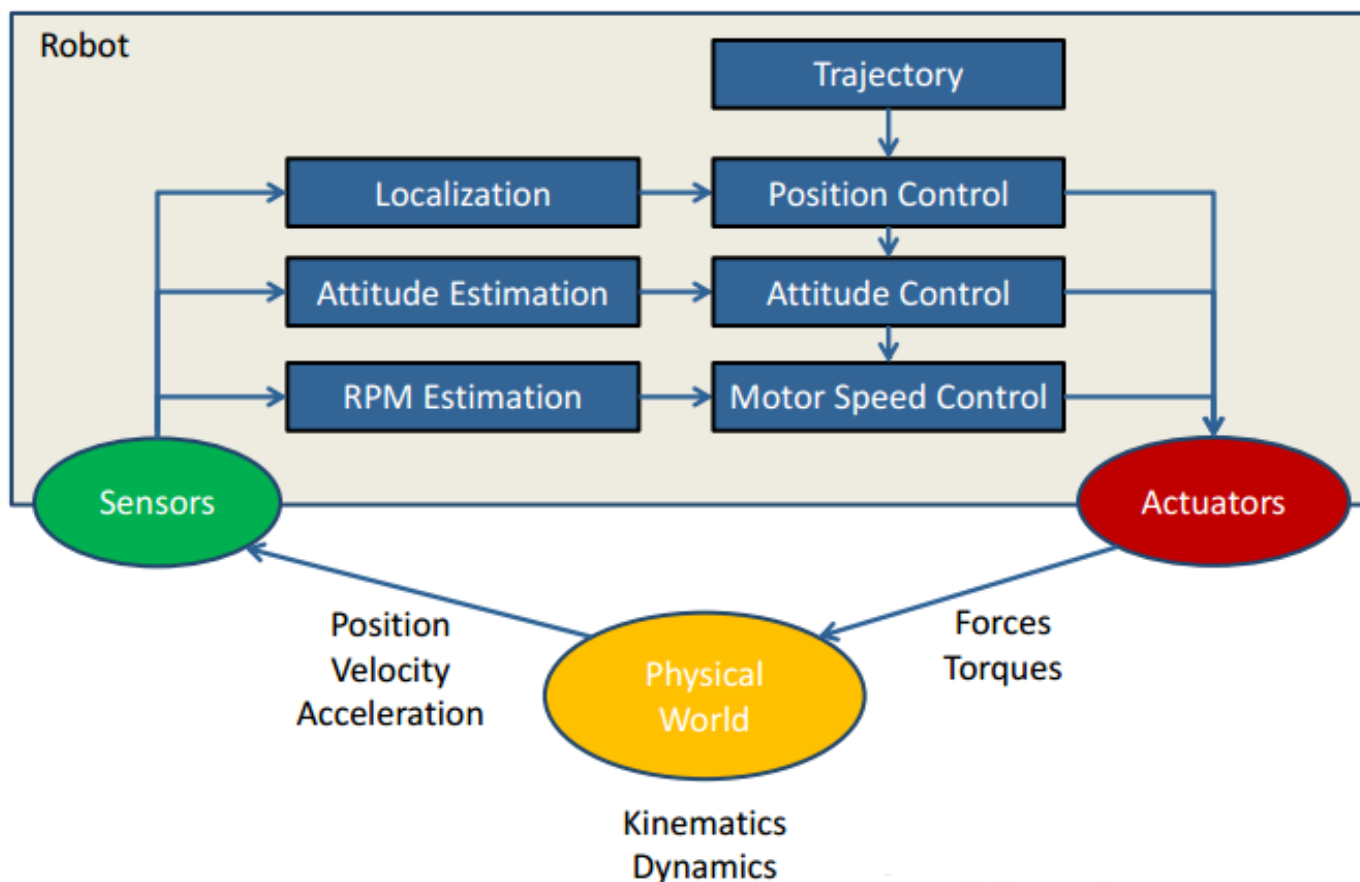


图2 基于速度误差校正的控制系统

Control Architecture

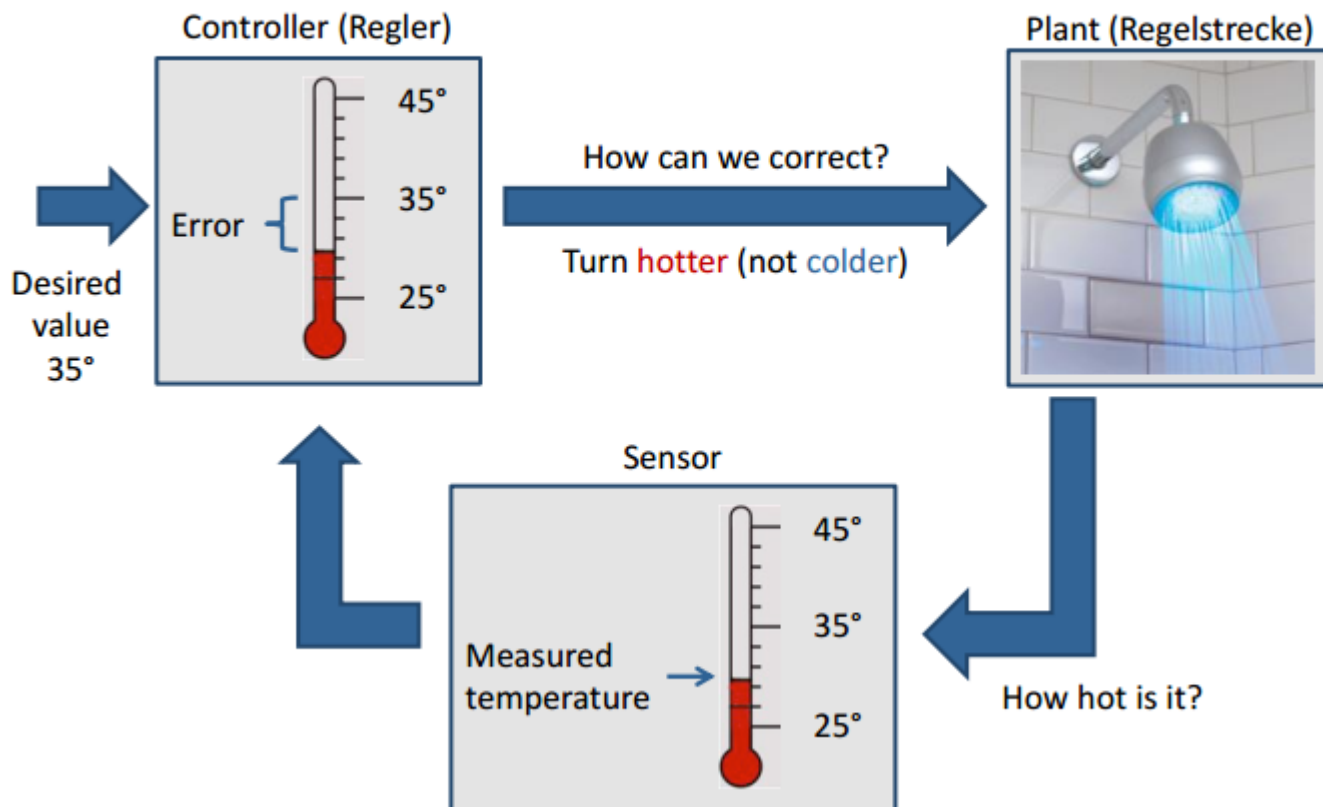




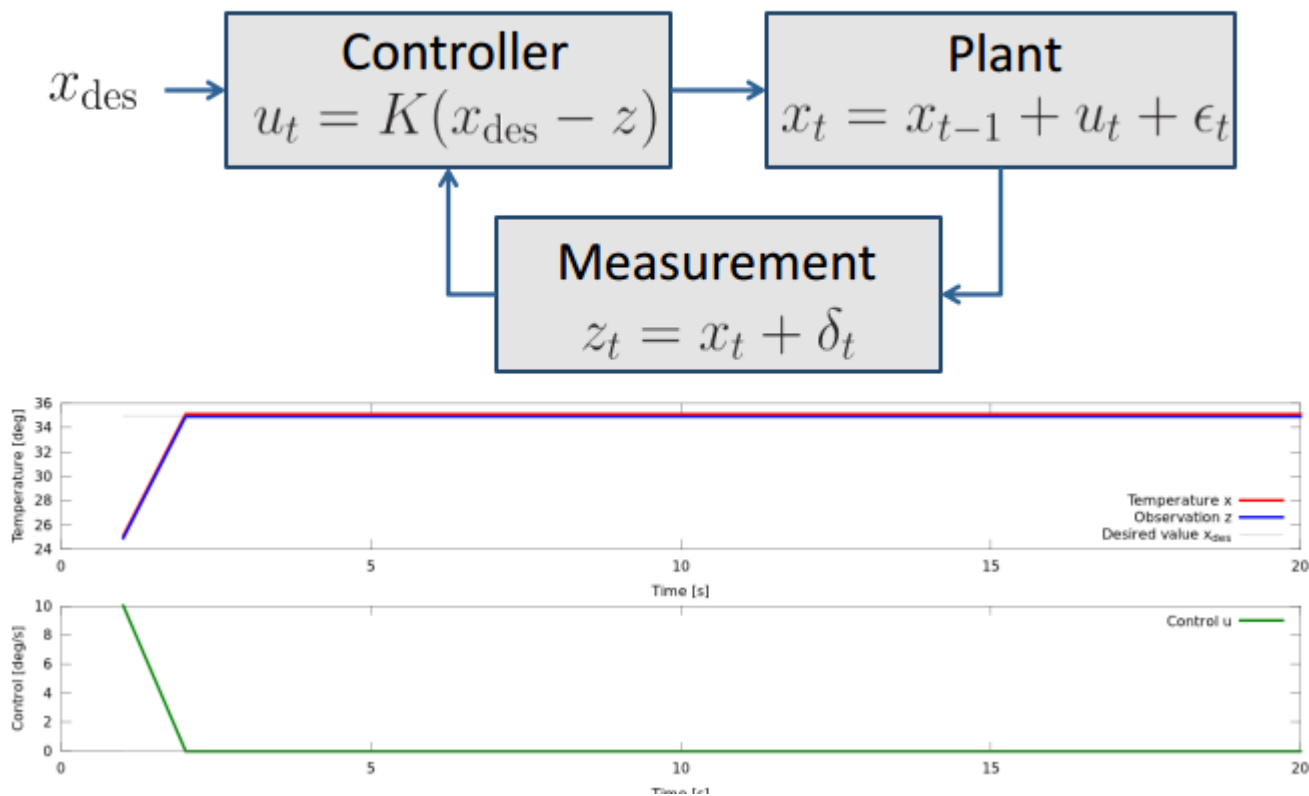
Feedback Control

- Given:
 - Goal state x_{des}
 - Measured state (feedback) z
- Wanted:
 - Control signal u to reach goal state
- How to compute the control signal?

Feedback Control - Generic Idea

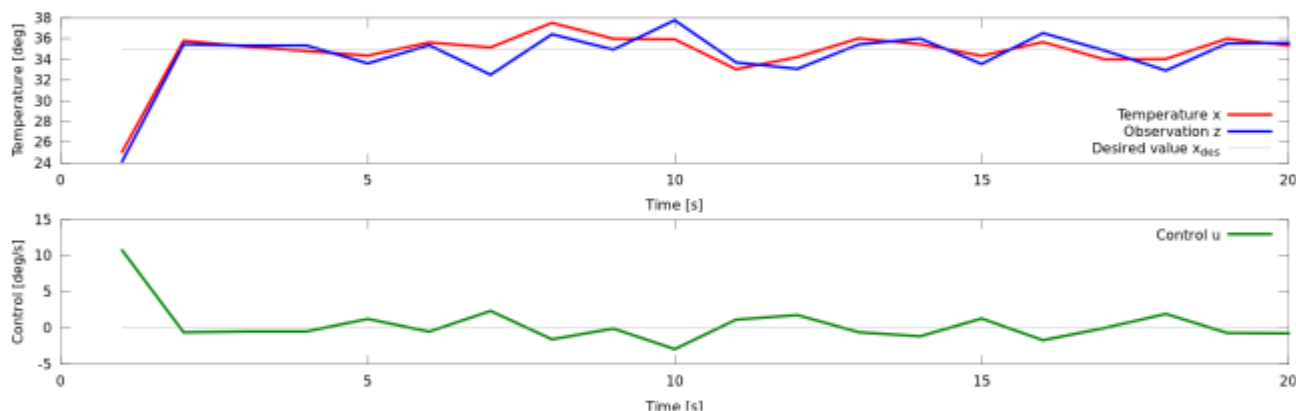


Feedback Control - Example



Measurement Noise

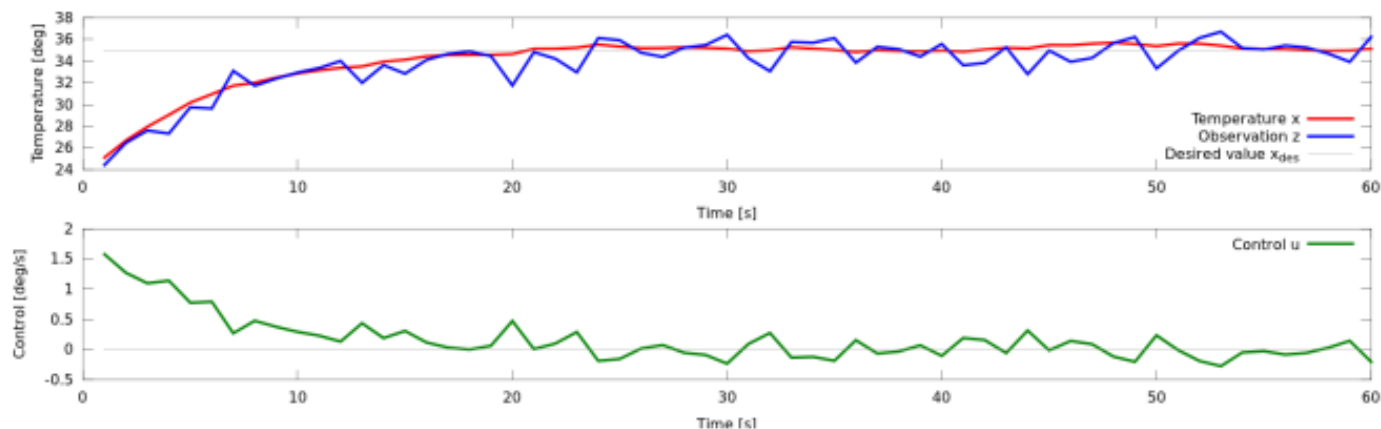
- What effect has noise in the measurements?



- Poor performance for $K=1$
- How can we fix this?

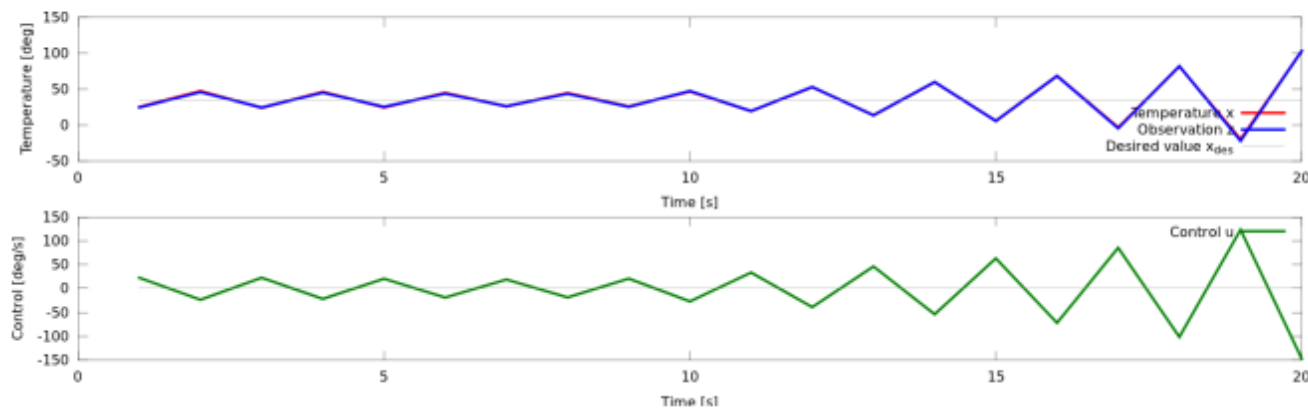
Proper Control with Measurement Noise

- Lower the gain... ($K=0.15$)



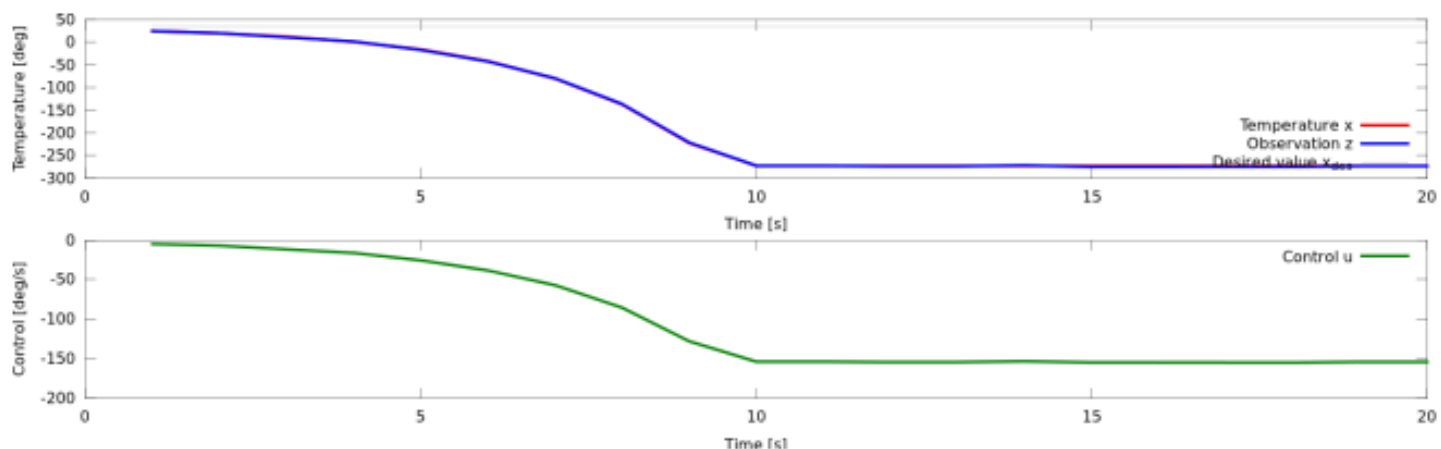
What do High Gains do?

- High gains are always problematic ($K=2.15$)



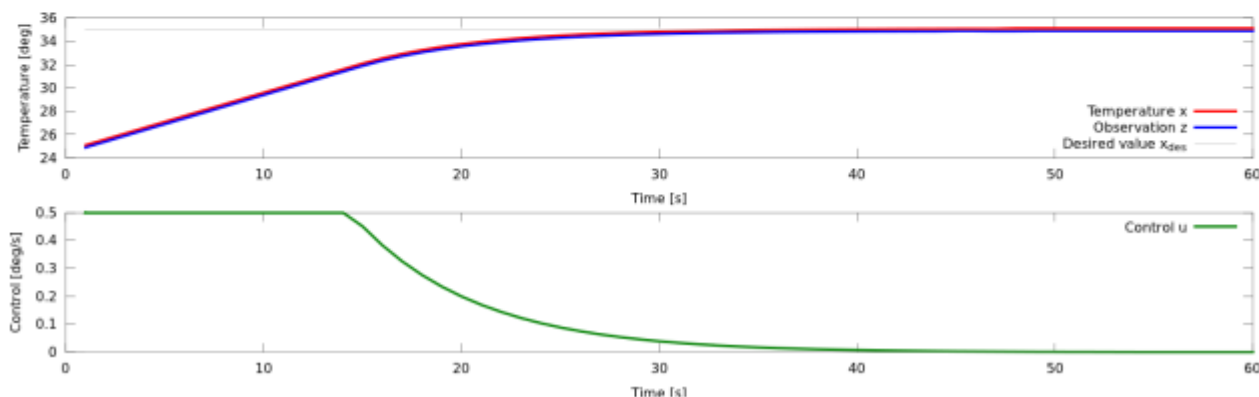
What happens if sign is messed up?

- Check $K=-0.5$

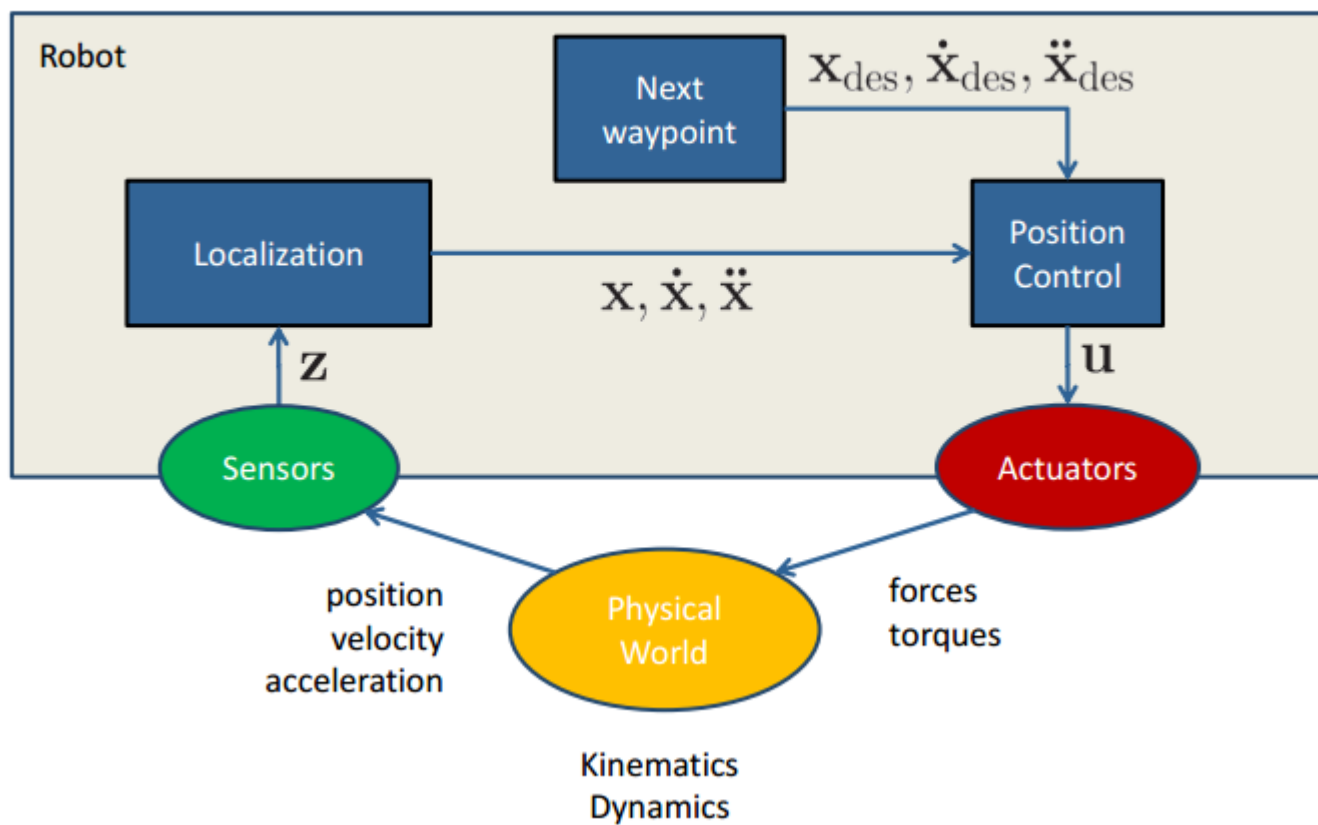


Saturation

- In practice, often the set of admissible controls u is bounded
- This is called (control) saturation

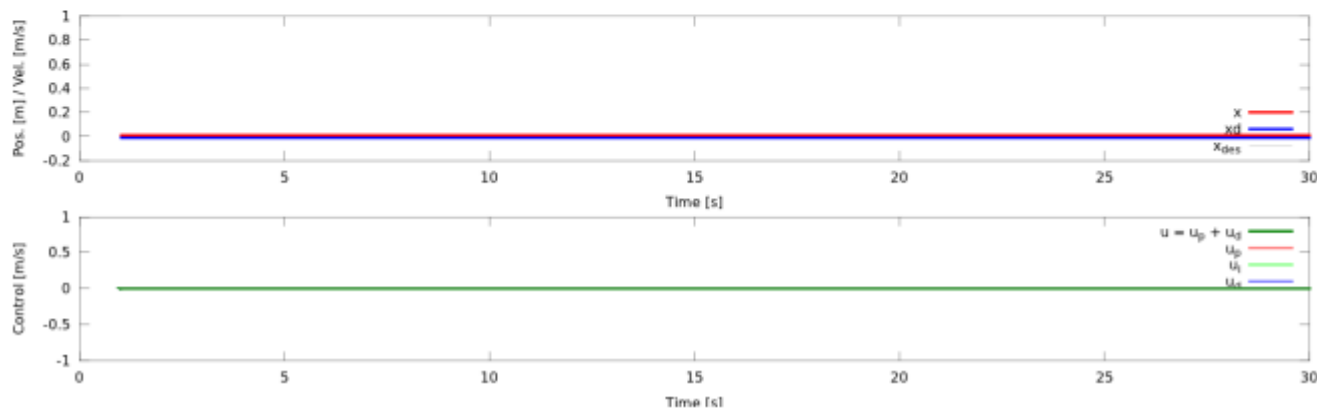


Position Control



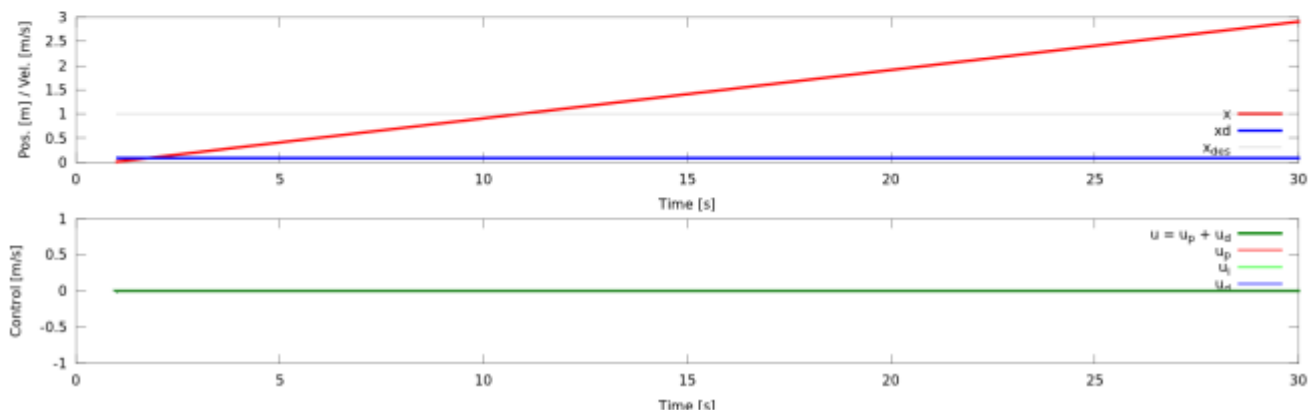
Rigid Body Kinematics

- Consider a rigid body
- Free floating in 1D space, no gravity
- How does this system evolve over time?
- Example: $x_0 = 0, \dot{x}_0 = 0$



Rigid Body Kinematics

- Consider a rigid body
- Free floating in 1D space, no gravity
- How does this system evolve over time?
- Example: $x_0 = 0, \dot{x}_0 = 0.1$





Rigid Body Kinematics

- Consider a rigid body
- Free floating in 1D space, no gravity
- In each time instant, we can apply a force F
- Results in acceleration $\ddot{x} = F/m$
- Desired position $x_{\text{des}} = 1$

P Control

- What happens for this control law?

$$u_t = K(x_{\text{des}} - x_{t-1})$$

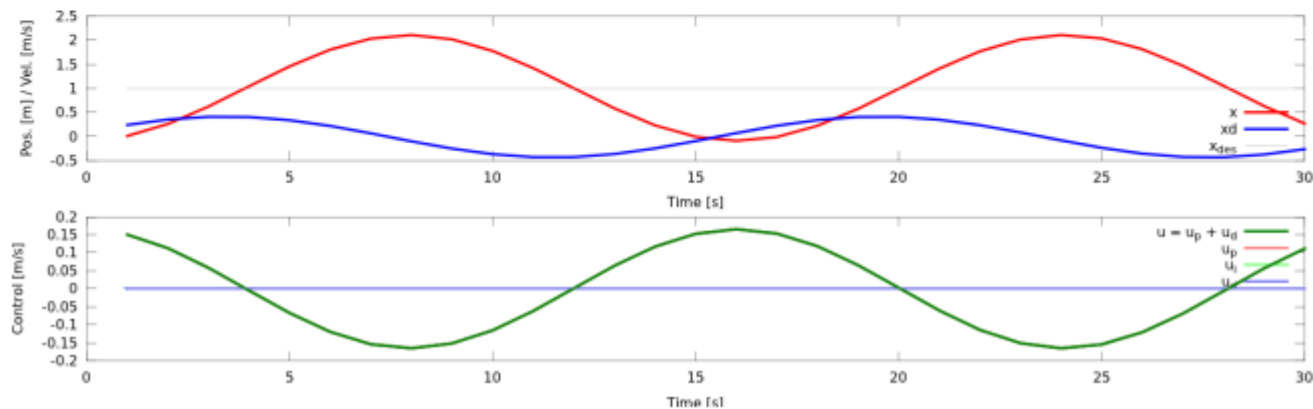
- This is called proportional control

P Control

- What happens for this control law?

$$u_t = K(x_{\text{des}} - x_{t-1})$$

- This is called proportional control

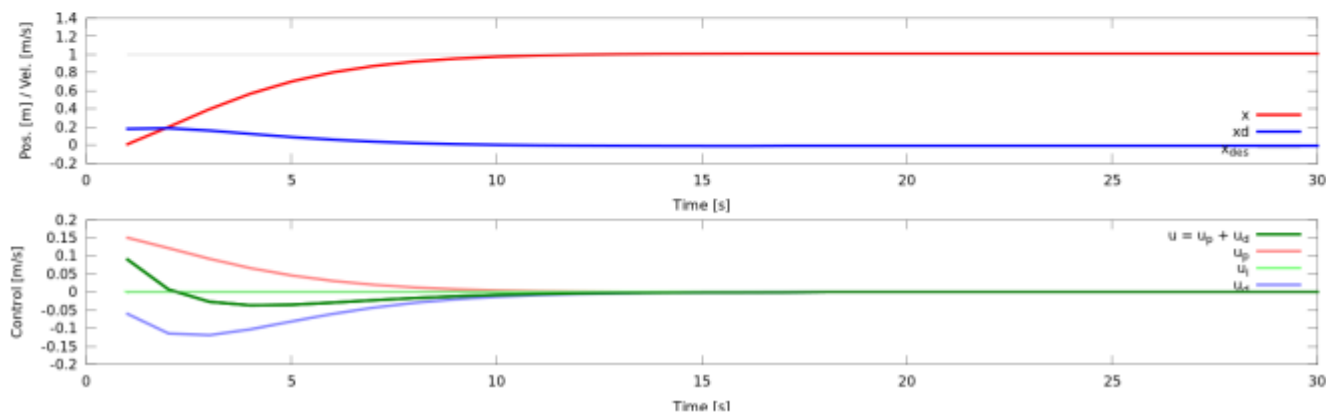


PD Control

- What happens for this control law?

$$u_t = K_P(x_{\text{des}} - x_{t-1}) + K_D(\dot{x}_{\text{des}} - \dot{x}_{t-1})$$

- Proportional-Derivative control

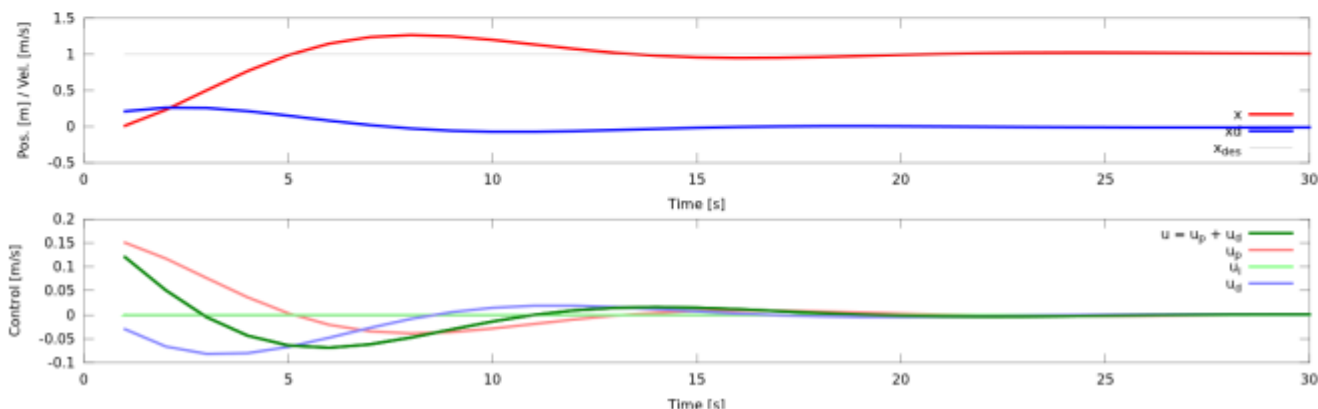


PD Control

- What happens for this control law?

$$u_t = K_P(x_{\text{des}} - x_{t-1}) + K_D(\dot{x}_{\text{des}} - \dot{x}_{t-1})$$

- What if we set **higher** gains?

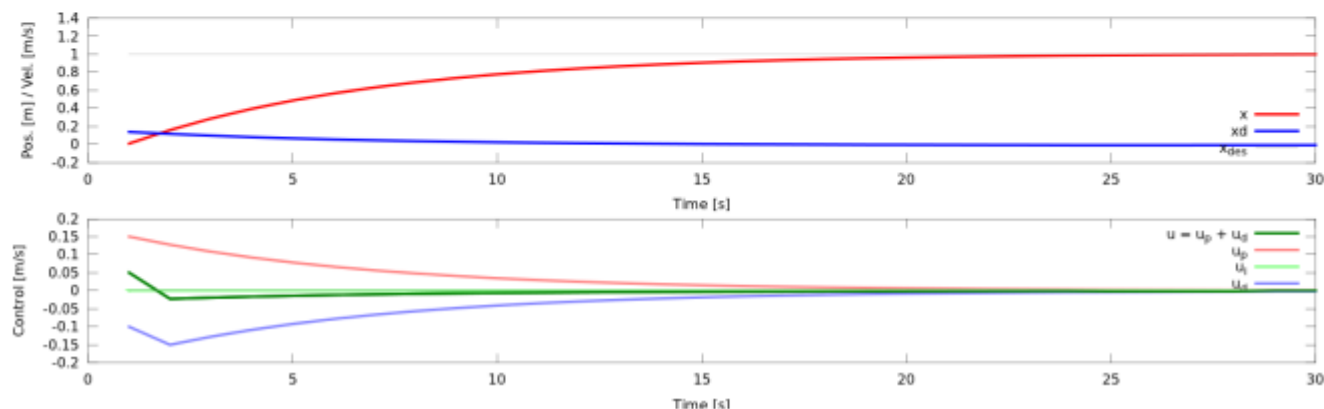


PD Control

- What happens for this control law?

$$u_t = K_P(x_{\text{des}} - x_{t-1}) + K_D(\dot{x}_{\text{des}} - \dot{x}_{t-1})$$

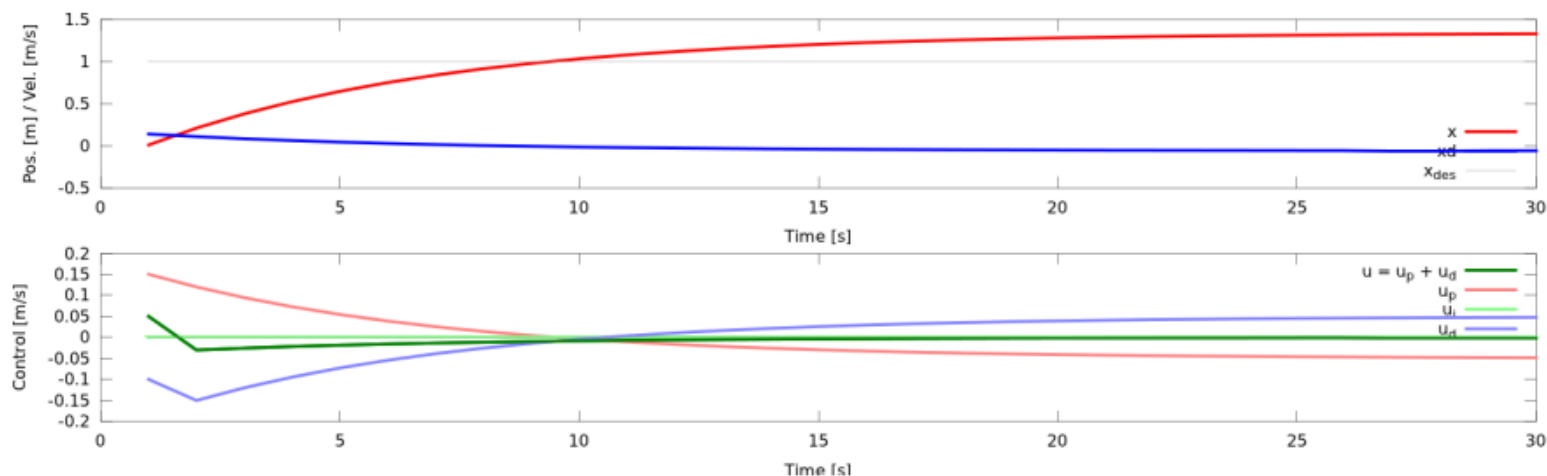
- What if we set **lower** gains?





PD Control

- What happens when we have systematic errors?
(control/sensor noise with non-zero mean)
- Example: unbalanced quadcopter, wind, ...
- Does the robot ever reach its desired location?

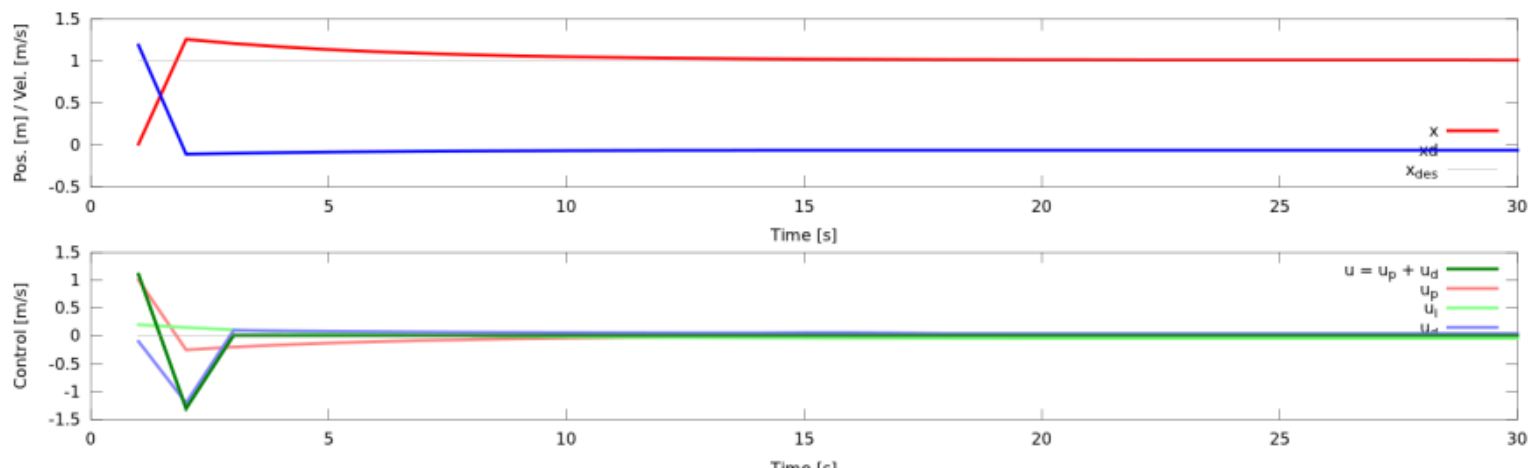


PID Control

- Idea: Estimate the system error (bias) by integrating the error

$$u_t = K_P(x_{\text{des}} - x_t) + K_D(\dot{x}_{\text{des}} - \dot{x}_t) + K_I \int_{-\infty}^t x_{\text{des}} - x_t dt$$

- Proportional+Derivative+Integral Control





PID Control

- Idea: Estimate the system error (bias) by integrating the error

$$u_t = K_P(x_{\text{des}} - x_t) + K_D(\dot{x}_{\text{des}} - \dot{x}_t) + K_I \int_{-\infty}^t x_{\text{des}} - x_t dt$$

- Proportional+Derivative+Integral Control
- For steady state systems, this can be reasonable
- Otherwise, it may create havoc or even disaster (wind-up effect)

Wind-up effect 积分饱和：如果执行机构已经到极限位置，仍然不能消除偏差时，由于积分作用，尽管PID控制器的运算结果继续增大或减小，但执行机构已无相应的动作

- 将积分累积量限制上下限，避免积分累积量超过限制值



PID Control – Summary

- P控制，比例控制：控制器的输出与误差信号成比例关系
 - 当仅有比例控制时系统输出存在稳态误差
 - 增大比例增益，一般会加快系统响应
 - 如果比例增益太大，过程变量会有振荡，如果继续增加增益，系统振荡会越来越大，以至于超出控制，使得系统变得不稳定

$$u_t = K_P(x_{\text{des}} - x_t) \cdot$$

稳态误差：稳定状态误差是过程变量和设定值之间的差值

有差系统：系统如果在进入稳态后存在稳态误差，则称这个控制系统是有稳态误差的或简称有差系统（System with Steady-state Error）



PID Control – Summary

□ 积分I控制：控制器的输出与输入误差信号的积分成正比关系

- ✓ 积分环节可以消除稳态误差
- ✓ 积分模块将一段时间内的误差相加，随着时间的增加，积分项会增大，因此即使误差很小，积分项也会随着时间的增加而加大，它推动控制器的输出增大使稳态误差进一步减小，直到等于零，达到无差调节的效果
- ✓ 比例+积分（PI）控制器，可以使稳态系统进入稳态后无稳态误差

$$u_t = K_P(x_{des} - x_t) + K_I \int_{-\infty}^t x_{des} - x_t dt$$

Wind-up effect 积分饱和：如果执行机构已经到极限位置，仍然不能消除偏差时，由于积分作用，尽管PID控制器的运算结果继续增大或减小，但执行机构已无相应的动作

- 将积分累积量限制上下限，避免积分累积量超过限制值

PID Control – Summary

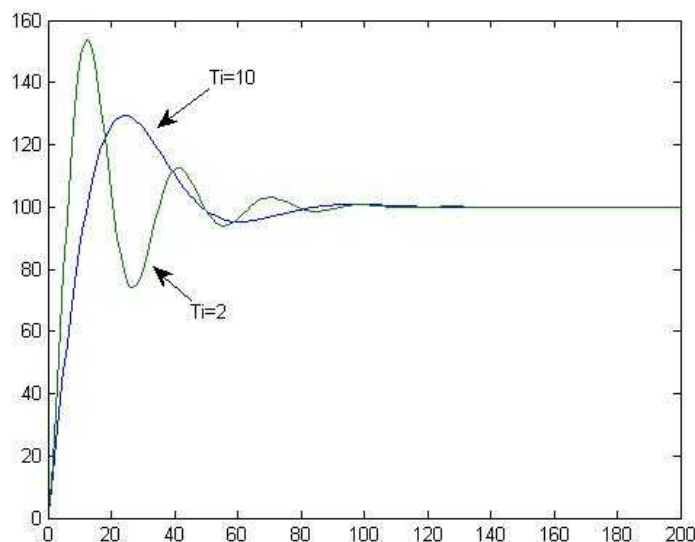
- 比例+积分，PI控制：控制器的输出与输入误差信号的积分成正比关系

$$u_t = K_P(x_{des} - x_t) + K_I \int_{-\infty}^t x_{des} - x_t dt$$

积分时间常数 T_I ：表示积分环节敏感度

$$K_I = K_p / T_I$$

- 增大积分时间常数 T_I 有利于减小超调，减小振荡，使系统的稳定性增加，但是系统稳态误差消除时间变长





PID Control – Summary

□ 微分控制D：控制器的输出与误差的变化率成正比

- ✓ 在克服误差的调节过程中，自动控制系统可能出现振荡甚至失稳，其原因主要是由于存在有较大惯性组件（环节）或有滞后组件，其变化总是落后于误差的变化
- ✓ 解决办法：使抑制误差作用的变化“超前”，即在误差接近零时，抑制误差的作用就应该是零。这就是说，在控制器中仅引入“比例P”项往往是不够的，比例项的作用仅是放大误差的幅值，增加“微分项”能预测误差变化的趋势
- ✓ 具有比例+微分的控制器，能提前使抑制误差的控制作用等于零，甚至为负值，从而避免了被控量的严重超调
- ✓ 对有较大惯性或滞后的被控对象，比例P+微分D（PD）控制器能改善系统在调节过程中的动态特性

$$u_t = K_P(x_{\text{des}} - x_t) + K_D(\dot{x}_{\text{des}} - \dot{x}_t)$$



PID Control – Summary

$$u_t = K_P(x_{\text{des}} - x_t) + K_D(\dot{x}_{\text{des}} - \dot{x}_t) + K_I \int_{-\infty}^t x_{\text{des}} - x_t dt$$

现在：比例环节P反映输入误差信号当前的信息

过去：积分环节I反映输入误差信号的“历史变化”

将来：微分环节D反映输入误差信号未来的变化趋势

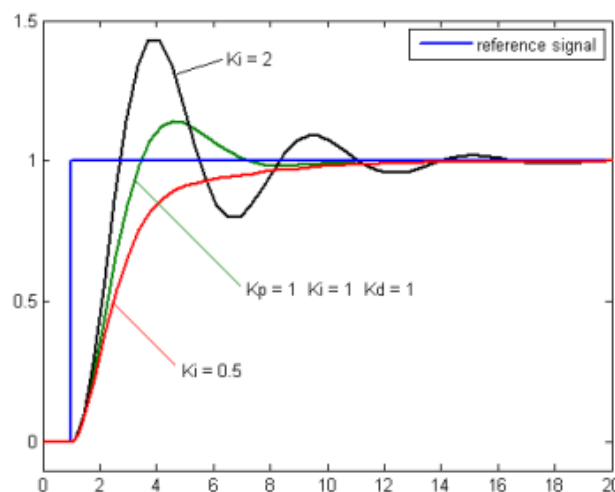
比例环节-矫正：根据当前的误差修正行为

积分环节-学习：根据过去的经验找到优解

微分环节-预测：根据未来的趋势做出调整

How to Choose the Coefficients?

- Gains too large: overshooting, oscillations
- Gains too small: long time to converge
- Heuristic methods exist
- In practice, often tuned manually





PID控制器参数整定

- 设置P、I、D最佳增益，从而得到控制系统理想反馈的过程
- PID参数整定有两大类：

理论计算整定法

- ✓ 依据系统的数学模型，经过理论计算确定控制器参数
- ✓ 计算得到的数据未必可以直接用，还需通过工程实际进行调整和修改

工程整定方法

- ✓ 依赖工程经验，直接在控制系统的试验中进行，且方法简单、易于掌握，在工程实际中被广泛采用
- ✓ 主要有临界比例法、反应曲线法和衰减法。三种方法各有其特点，其共同点都是通过试验，然后按照工程经验公式对控制器参数进行整定
- ✓ 无论采用哪种方法所得到的控制器参数，都需要在实际运行中进行最后调整与完善



PID控制器参数整定

➤ 工程整定方法—临界比例法

利用该方法进行 PID控制器参数的整定步骤如下：

- (1) 首先预选择一个足够短的采样周期让系统工作；
- (2) 仅加入比例控制环节，直到系统对输入的阶跃响应出现临界振荡，记下这时的比例放大系数和临界振荡周期；
- (3) 在一定的控制度下通过公式计算得到PID控制器的参数

机器人的控制：PID控制



PID参数怎样调整最佳

(1) 整定比例控制

将比例控制作用由小变到大，观察各次响应，直至得到反应快、超调小的响应曲线。

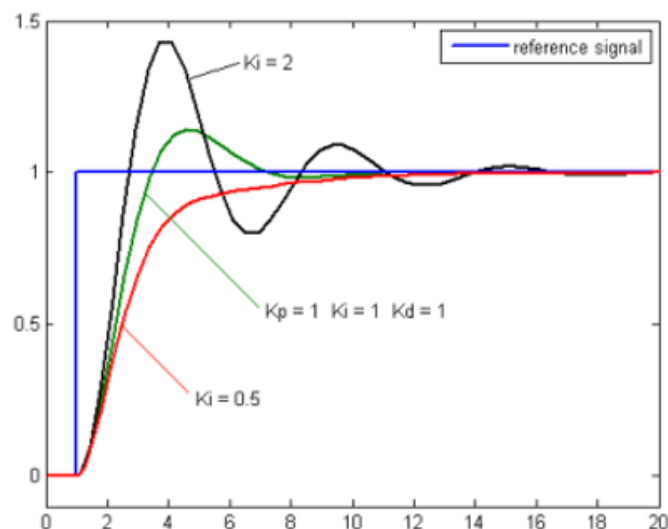
(2) 整定积分环节

若在比例控制下稳态误差不能满足要求，需加入积分控制。

先将步骤(1)中选择的比例系数减小为原来的50~80%，再将积分时间置一个较大值，观测响应曲线。然后减小积分时间，加大积分作用，并相应调整比例系数，反复试凑至得到较满意的响应，确定比例和积分的参数。

(3) 整定微分环节

若经过步骤(2)，PI控制只能消除稳态误差，而动态过程不能令人满意，则应加入微分控制，构成PID控制。先置微分时间 $TD=0$ ，逐渐加大 TD ，同时相应地改变比例系数和积分时间，反复试凑至获得满意的控制效果和PID控制参数。





离散PID控制器

增量式PID控制器

自整定PID控制器

...



极点配置：

- 给定控制系统，通过设计反馈增益 k 使闭环系统具有期望的极点，从而达到适当的阻尼系数和无阻尼自然频率
- 极点配置是基于状态反馈，即 $u=-kx$ ，因此状态 x 需可测
- 当状态不可测时，则应设计状态观测器
- 设计的状态观测器也应具有适当的频率特性，因此也可指定其极点的位置，从而使状态观测器的设计转化为极点配置问题

MatLab Control Toolbox: 极点配置和状态估计器

函数名称和典型输入变元	功 能
acker(A,B,p)	SISO系统极点配置
place(A,B,p)	MIMO系统极点配置
estim(sys,L)	生成系统状态估计器
reg(sys,K,L)	生成系统调节器

机器人的控制：LQR控制



对于线性时不变 (LTI) 系统：

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$y(t) = Cx(t) + Du(t)$$

线性二次型 (LQ) 最优控制器的任务是设计 $u(t)$ ，使线性二次型最优控制指标 (代价函数) 最小：

$$J = \frac{1}{2} \int_{t_0}^{t_f} (x^T Q x + u^T R u + x^T N u) dt$$

- 线性二次型的性能指标易于分析、计算、处理，在控制系统的各个领域内都得到了广泛地重视和应用
- 线性二次型调节器 (LQR) 控制的闭环系统，其动态响应与加权系数矩阵 Q 和 R 之间存在着复杂的对应关系，这给加权矩阵的选择带来许多困难，目前普遍采用的仿真试凑法获得的最优控制是“人工”意义下的最优。在实际中，采用基于仿真的优化方法求取最满意解是解决这类问题的最佳途径

MatLab Control Toolbox: LQ最优控制函数

函数名称和典型输入变元	功 能
<code>lqr(A,B,Q,R)</code>	连续系统的LQ调节器设计
<code>lqr2(A,B,Q,R)</code>	连续系统的LQ调节器设计
<code>dlqr(A,B,Q,R)</code>	离散系统的LQ调节器设计
<code>lqry(A,B,Q,R)</code>	系统的LQ调节器设计
<code>lqrd(A,B,Q,R,Ts)</code>	连续代价函数的离散LQ调节器设计
<code>kalman(sts,Qn,Rn,Nn)</code>	系统的Kalman滤波器设计
<code>kalmd(sys,Qn,Rn,Ts)</code>	连续系统的离散Kalman滤波器设计
<code>lqgreg(kest,k)</code>	根据Kalman和状态反馈增益设计调节器

机器人的控制：控制系统的性能指标



控制系统的性能指标

控制系统性能评价分为**动态性能指标**和**稳态性能指标**两类。为了求解系统的时间响应，必须了解输入信号（即外作用）的解析表达式

典型输入信号

- 研究分析系统的性能指标时，通常选择若干典型输入信号作为动态性能指标和稳态性能测试之用，典型输入信号包括：

单位阶跃函数、单位斜坡函数、单位加速度函数、单位脉冲函数、正弦函数

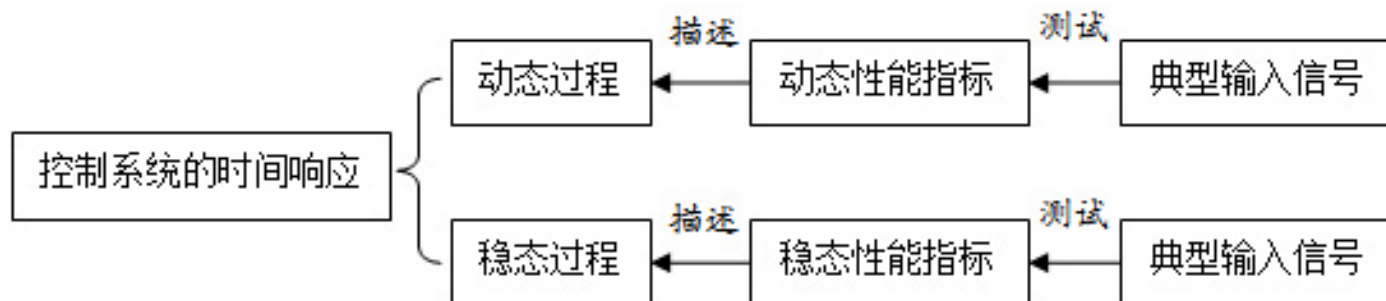
- 实际中采用上述哪种典型输入信号取决于系统常见的工作状态。同时，在所有可能输入信号中，选择最不利的信号作为典型输入信号测试系统指标

机器人的控制：控制系统的性能指标



典型输入信号

- 在典型输入信号作用下，任何控制系统的时间响应都有动态过程和稳态过程两部分组成。因此，系统的时间响应、动态、稳态过程与动态、稳态性能指标以及典型输入信号的关系如下图所示。



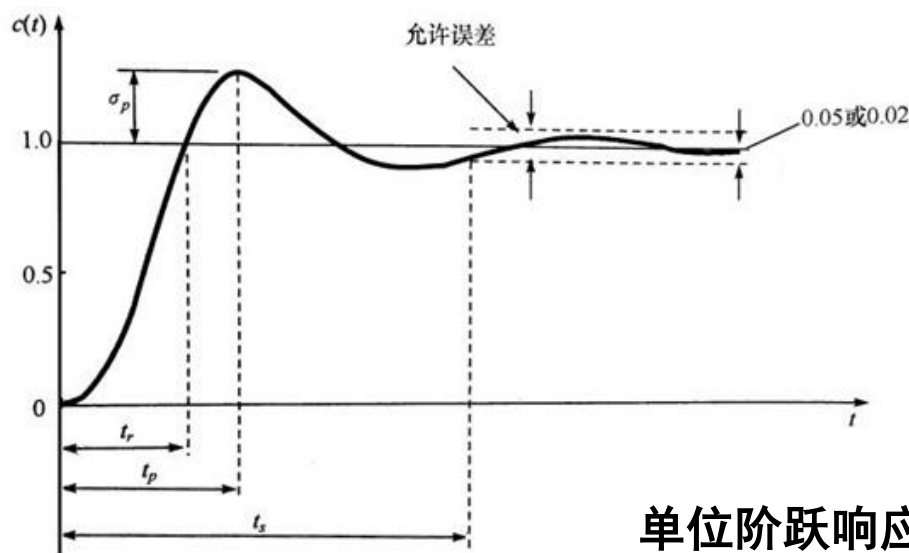
机器人的控制：控制系统的性能指标



动态过程与动态性能指标

动态过程

动态过程又称**过渡过程**或**瞬态过程**，指系统在典型输入信号作用下，系统输出量从初始状态到最终状态的响应过程。由于实际控制系统具有惯性、摩擦及其他原因，系统输出量不可能完全复现输入量的变化。根据系统结构和参数的选择，动态过程表现为衰减、发散、等幅振荡等形式。



单位阶跃响应曲线

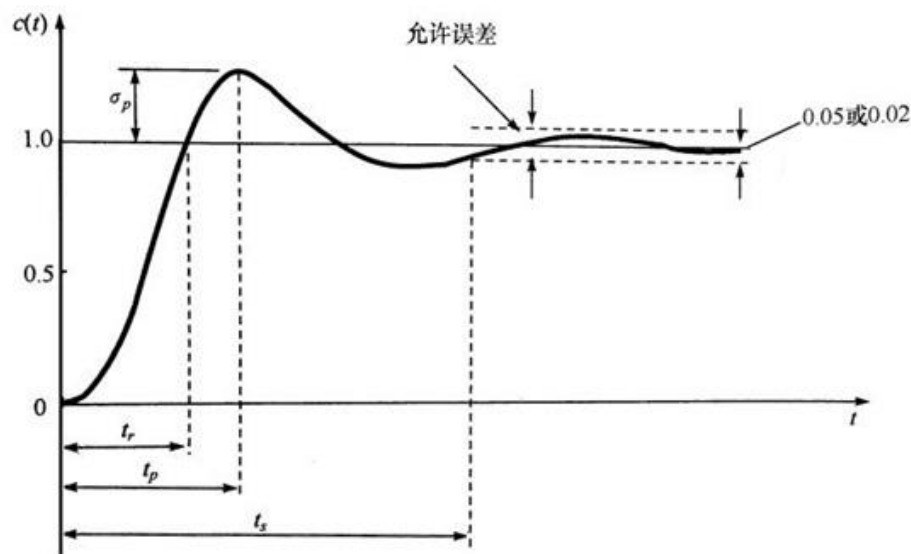
机器人的控制：控制系统的性能指标



动态过程与动态性能指标

动态性能

系统的动态过程提供系统稳定性、响应速度及阻尼情况，由动态性能指标描述。通常在阶跃函数作用下，测定或计算系统的动态性能。描述稳定的系统在单位阶跃函数的作用下，动态过程随时间 t 的变化状况的指标，称为动态性能指标。系统的单位阶跃响应如下图所示。



机器人的控制：控制系统的性能指标



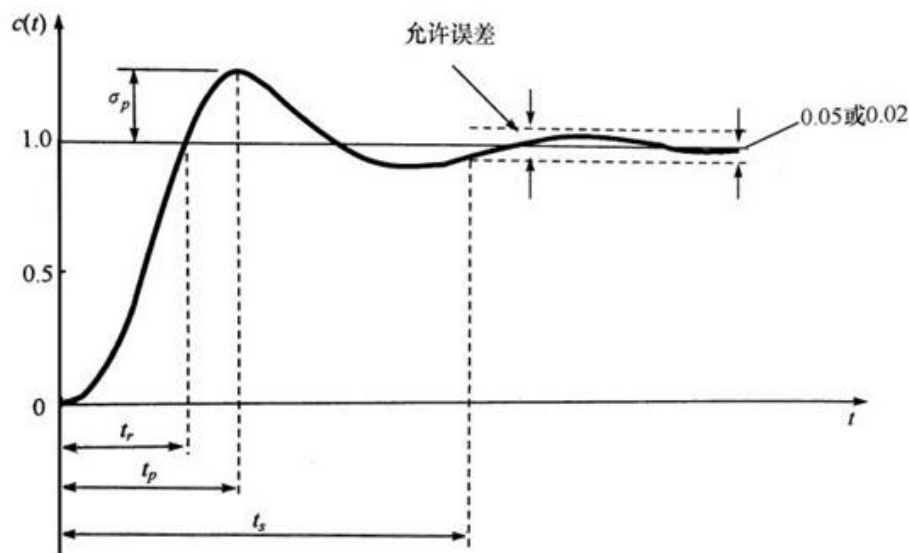
动态过程与动态性能指标

- 上升时间 (rise time) t_r

指响应从终值10%上升到终值90%所需的时间；对于有振荡的系统，也可定义为响应从零第一次上升到终值所需的时间。**上升时间 t_r 是系统响应速度的一种度量。**上升时间越短，响应速度越快。

- 峰值时间 t_p

指响应超过其终值达到第一个峰值所需的时间。**上升时间 t_p 是系统响应速度的一种度量。**



机器人的控制：控制系统的性能指标



动态过程与动态性能指标

- 调节时间 (setting time) t_s

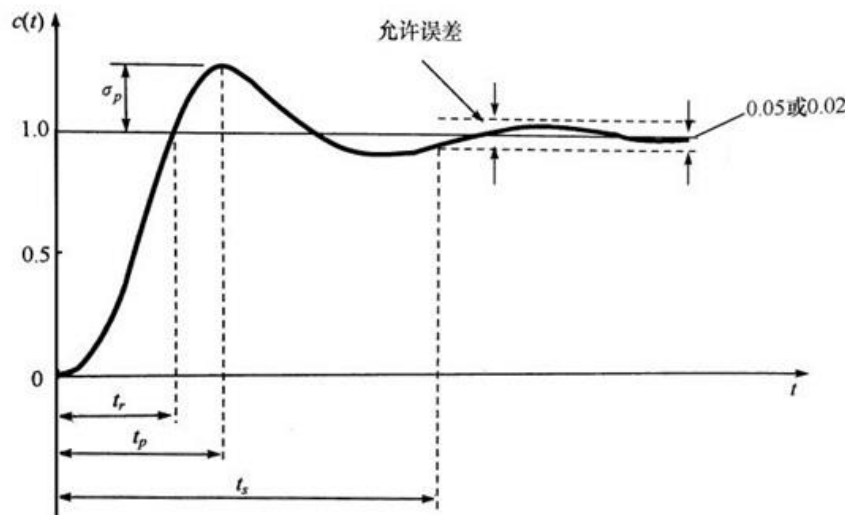
指响应到达并保持在终值 $\pm 5\%$ (或 $\pm 2\%$) 内所需的最短时间。调节时间 t_s 是评价系统响应速度和阻尼程度的综合指标。

- 超调量 (overshoot) : σ %

指响应的最大偏离量 $c(t_p)$ 与终值 $c(\infty)$ 之比的百分数，即

$$\sigma\% = \frac{c(t_p) - c(\infty)}{c(\infty)} \times 100\%$$

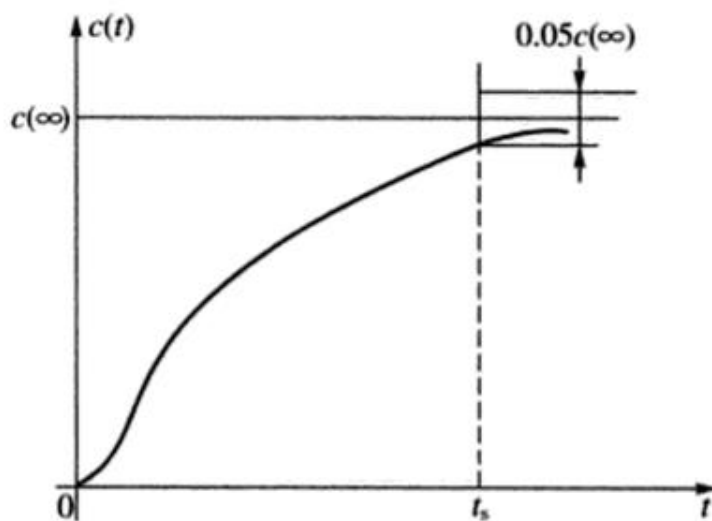
若 $c(t_p) < c(\infty)$ ，则响应无超调。 σ %评价系统的阻尼程度。



稳态过程与稳态性能指标

➤ 稳态过程

稳态过程指系统在典型输入信号作用下，当时间 t 趋于无穷时，系统输出量的表现方式。稳态过程又称为稳态响应，表征系统输出量最终复现输入量的程度，提供系统有关稳态误差的信息，用稳态性能描述



单位阶跃响应曲线



稳态过程与稳态性能指标

➤ 稳态性能—稳态误差 (steady-state error)

- ✓ 稳态误差是描述系统稳态性能的一种性能指标，通常在阶跃函数、斜坡函数或加速度函数作用下进行测定或计算。若时间区域无穷，系统的输出量不等于输入量或者输入量的确定函数，则系统存在稳态误差
- ✓ 稳态误差是系统控制精度或抗扰动能力的一种度量
- ✓ 稳态误差分为原理性误差和实际性误差
 - 原理性误差：由于外扰动作用的存在，控制系统在原理上必然存在的一类稳态误差。当原理性稳态误差为零时，控制系统称为无静差系统，否则称为有静差系统
 - 实际性误差：系统的组成部件中的不完善因素（如摩擦、间隙、不灵敏区等）所造成的稳态误差，实际性误差不可能完全消除，只能通过选用高精度的部件等途径减小

著名机器人实验室与研究团队



<http://www.iris.ethz.ch/>

<http://ruina.tam.cornell.edu/>

<https://www.grasp.upenn.edu/research>

<http://www.iris.ethz.ch/>

<http://www.asl.ethz.ch/>

<https://www.eng.yale.edu/grablab/>

<https://groups.csail.mit.edu/locomotion/>

<http://automation.berkeley.edu/index.html>

<https://people.eecs.berkeley.edu/~pabbeel/>

<http://homes.cs.washington.edu/~todorov/>

<http://pr.cs.cornell.edu/>

<http://www-cvr.ai.uiuc.edu/~seth/res.php?u=home>