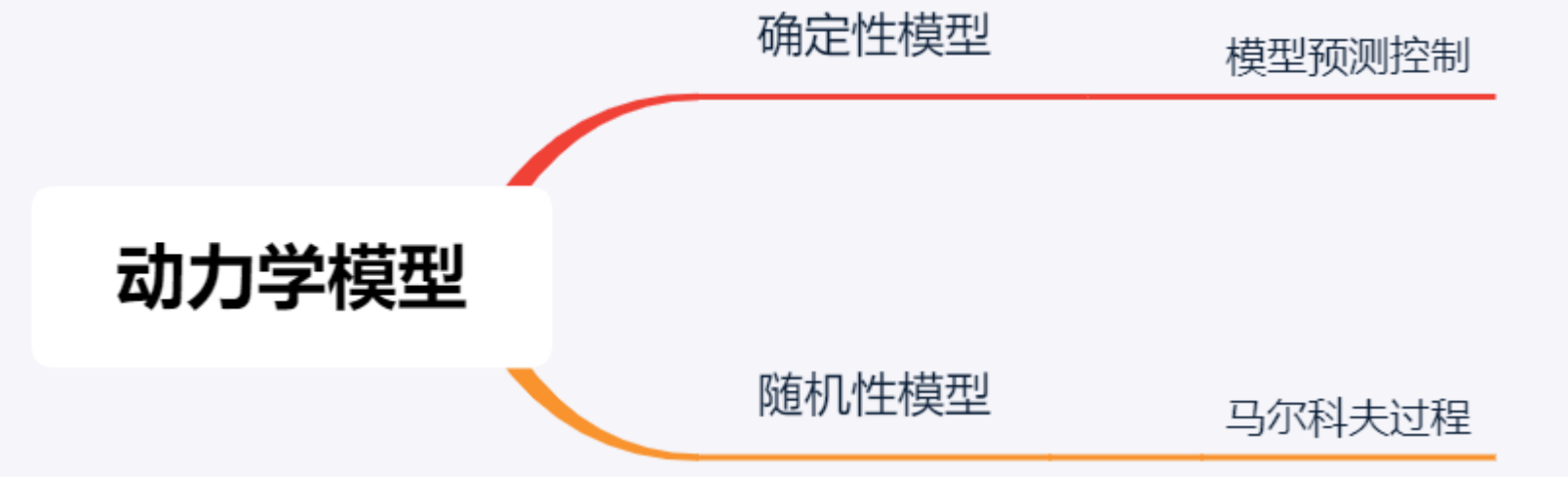
11 模型预测控制

- **动力学模型**

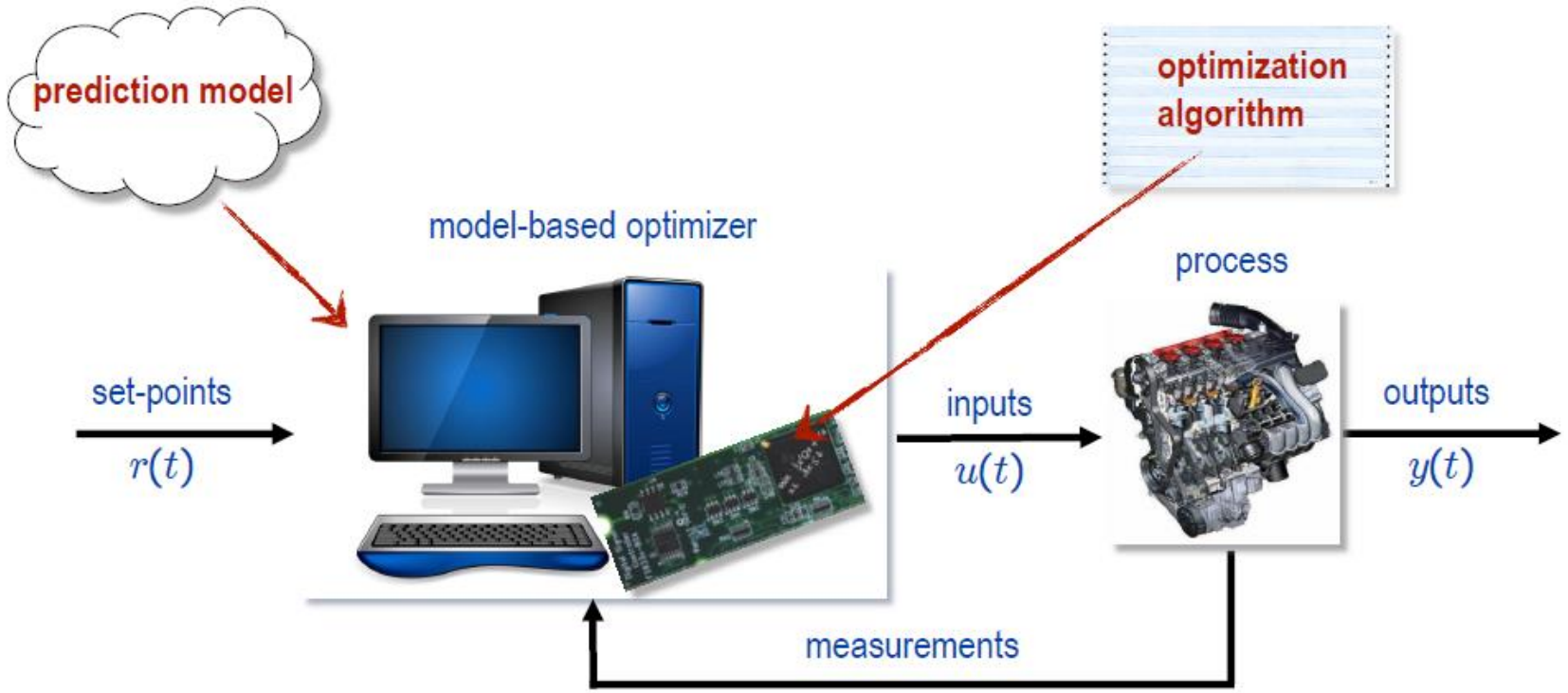
  - 从系统的角度分析：动力学表达了系统演化的动态规律
  - 从统计意义来看：模型分为随机模型和确定性模型



11 模型预测控制

- **模型预测控制概念**

  - 动力学模型：简化 VS 完整
  - 演化过程：有限 VS 无限
  - 控制策略：有效 VS 最佳



simplified　　likely

Use a dynamical model of the process to predict its future evolution and choose ~~the "best" control~~ a good action

11 模型预测控制

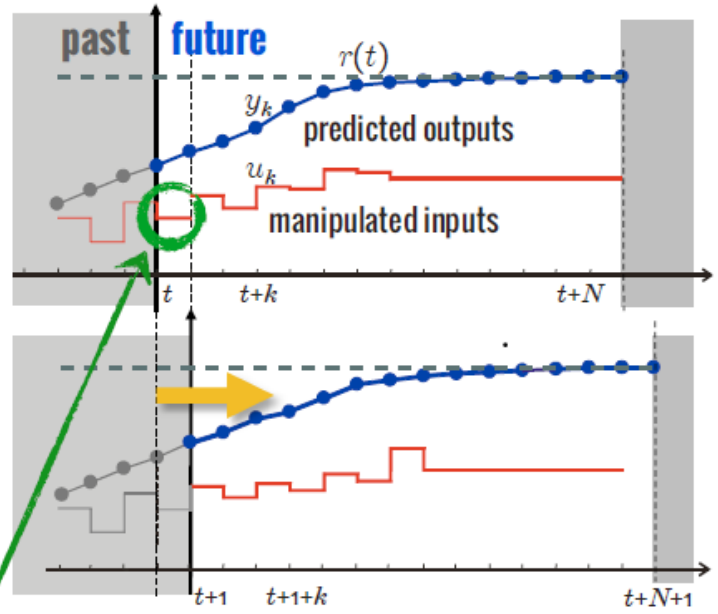
- **模型预测控制求解**
  - 多步优化
  - 单步控制

- **Goal**: find the best control sequence over a future horizon of N steps

$$\min \sum_{k=0}^{N-1} \|W^y(y_k - r(t))\|_2^2 + \|W^u(u_k - u_r(t))\|_2^2$$

$$\text{s.t.} \quad x_{k+1} = f(x_k, u_k) \qquad \text{prediction model}$$
$$y_k = g(x_k)$$
$$u_{\min} \le u_k \le u_{\max} \qquad \text{constraints}$$
$$y_{\min} \le y_k \le y_{\max}$$
$$x_0 = x(t) \qquad \text{state feedback}$$

→ **numerical optimization problem**



- **At each time** t:
  - get new measurements to update the estimate of the current state $x(t)$
  - solve the optimization problem with respect to $\{u_0, \ldots, u_{N-1}\}$
  - apply only the first optimal move $u(t) = u_0^*$, discard the remaining samples

- 重点讨论
  - **无约束线性MPC问题**
  - **有约束线性MPC问题**

Bemporad. Model Preditive Control. 2020.

- **线性MPC——无约束问题(状态镇定)**

  - LQR问题的MPC提法
  - **状态镇定问题**

- Linear prediction model

$$\begin{cases} x_{k+1} &= Ax_k + Bu_k \\ y_k &= Cx_k \end{cases}$$

$$x \in \mathbb{R}^n$$
$$u \in \mathbb{R}^m$$
$$y \in \mathbb{R}^p$$

Notation:

$x_0 = x(t)$

$x_k = x(t+k|t)$

$u_k = u(t+k|t)$

- Relation between input and states: $x_k = A^k x_0 + \sum_{j=0}^{k-1} A^j B u_{k-1-j}$

- Performance index

$$J(z, x_0) = x'_N P x_N + \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k$$

$$R = R' \succ 0 \qquad Q = Q' \succeq 0 \qquad P = P' \succeq 0$$

$$z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

- **Goal**: find the sequence z^* that minimizes $J(z, x_0)$, i.e., that steers the state x to the origin optimally

- **线性MPC——无约束问题(状态镇定)**

  - MPC问题的推导

$$
\begin{aligned}
J(z,x_0) &= x_0'Qx_0 + \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix}' \overbrace{\begin{bmatrix} Q & 0 & 0 & \dots & 0 \\ 0 & Q & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & Q & 0 \\ 0 & 0 & \dots & 0 & P \end{bmatrix}}^{\bar{Q}} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} \\
&\quad + \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}' \underbrace{\begin{bmatrix} R & 0 & \dots & 0 \\ 0 & R & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & R \end{bmatrix}}_{\bar{R}} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}
\end{aligned}
$$

$$
\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \underbrace{\begin{bmatrix} B & 0 & \dots & 0 \\ AB & B & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \dots & B \end{bmatrix}}_{\bar{S}} \underbrace{\begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}}_{z} + \underbrace{\begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix}}_{\bar{T}} x_0
$$

$$
\begin{aligned}
J(z,x_0) &= (\bar{S}z + \bar{T}x_0)'\bar{Q}(\bar{S}z + \bar{T}x_0) + z'\bar{R}z + x_0'Qx_0 \\
&= \frac{1}{2}z' \underbrace{2(\bar{R} + \bar{S}'\bar{Q}\bar{S})}_{H} z + x_0' \underbrace{2\bar{T}'\bar{Q}\bar{S}}_{F'} z + \frac{1}{2}x_0' \underbrace{2(Q + \bar{T}'\bar{Q}\bar{T})}_{Y} x_0
\end{aligned}
$$

- **线性MPC——无约束问题(状态镇定)**

$$J(z,x_0)=\frac{1}{2}z'Hz+x_0'F'z+\frac{1}{2}x_0'Yx_0 \qquad z=\begin{bmatrix}u_0\\u_1\\\vdots\\u_{N-1}\end{bmatrix}$$

condensed form of MPC

  - 解的讨论
  - 梯度法求导，“批最优解”
  - 输出跟踪问题的Riccati方程迭代法
  - 等式约束问题的求解
  - 无约束QP问题的解

无约束线性MPC = 线性状态反馈

- The optimum is obtained by zeroing the gradient

$$\nabla_z J(z,x_0)=Hz+Fx_0=0$$

and hence $z^*=\begin{bmatrix}u_0^*\\u_1^*\\\vdots\\u_{N-1}^*\end{bmatrix}=-H^{-1}Fx_0$ (“batch” solution)

- Minimize quadratic function (no constraints)

$$\min_z f(z)=\frac{1}{2}z'Hz+x'(t)F'z \qquad z=\begin{bmatrix}u_0\\u_1\\\vdots\\u_{N-1}\end{bmatrix}$$

- solution: $\nabla f(z)=Hz+Fx(t)=0 \Rightarrow z^*=-H^{-1}Fx(t)$

$$\Rightarrow u(t)=-\begin{bmatrix}I & 0 & \dots & 0\end{bmatrix}H^{-1}Fx(t)=Kx(t)$$

unconstrained linear MPC = linear state-feedback!

- **线性MPC——有约束问题(状态镇定)**

  - LQR问题的MPC提法
  - **状态镇定问题**

- Linear prediction model:

$$\begin{cases} x_{k+1} = Ax_k + Bu_k \\ y_k = Cx_k \end{cases} \qquad \begin{matrix} x \in \mathbb{R}^n \\ u \in \mathbb{R}^m \\ y \in \mathbb{R}^p \end{matrix}$$

- Constraints to enforce:

$$\begin{cases} u_{\min} \le u(t) \le u_{\max} \\ y_{\min} \le y(t) \le y_{\max} \end{cases} \qquad \begin{matrix} u_{\min}, u_{\max} \in \mathbb{R}^m \\ y_{\min}, y_{\max} \in \mathbb{R}^p \end{matrix}$$

- Constrained optimal control problem (quadratic performance index):

$$\begin{aligned} \min_{z} \quad & x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \\ \text{s.t.} \quad & u_{\min} \le u_k \le u_{\max}, \; k = 0, \ldots, N-1 \\ & y_{\min} \le y_k \le y_{\max}, \; k = 1, \ldots, N \end{aligned}$$

$$\begin{matrix} R = R' \succ 0 \\ Q = Q' \succeq 0 \\ P = P' \succeq 0 \end{matrix} \qquad z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

- **线性MPC——有约束问题(状态镇定)**

  - MPC问题的推导

    - Input constraints $u_{\min} \le u_k \le u_{\max},\ k = 0, \dots, N-1$

$$\begin{cases} u_k \le u_{\max} \\ -u_k \le -u_{\min} \end{cases} \Rightarrow \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & 1 \\ -1 & 0 & \dots & 0 \\ 0 & -1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \dots & 0 & -1 \end{bmatrix} z \le \begin{bmatrix} u_{\max} \\ u_{\max} \\ \vdots \\ u_{\max} \\ -u_{\min} \\ -u_{\min} \\ \vdots \\ -u_{\min} \end{bmatrix} \qquad z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

    - Output constraints $y_k = CA^k x_0 + \sum_{i=0}^{k-1} CA^i B u_{k-1-i} \le y_{\max},\ k = 1, \dots, N$

$$\begin{bmatrix} CB & 0 & \dots & 0 \\ CAB & CB & \dots & 0 \\ \vdots & & & \vdots \\ CA^{N-1}B & \dots & CAB & CB \end{bmatrix} z \le \begin{bmatrix} y_{\max} \\ y_{\max} \\ \vdots \\ y_{\max} \end{bmatrix} - \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix} x_0$$

- **线性MPC——有约束问题(状态镇定)**

➢ MPC问题的推导

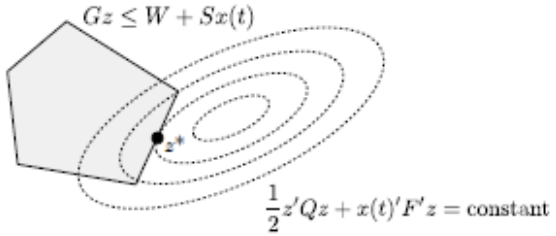
- Linear prediction model: $x_k = A^k x_0 + \sum_{i=0}^{k-1} A^i B u_{k-1-i}$
- Optimization problem (condensed form):

$$V(x_0) = \tfrac{1}{2} x_0' Y x_0 + \min_z \ \tfrac{1}{2} z' H z + x_0' F' z \quad \text{(quadratic objective)}$$

$$\text{s.t.} \quad Gz \leq W + S x_0 \quad \text{(linear constraints)}$$

convex Quadratic Program (QP)

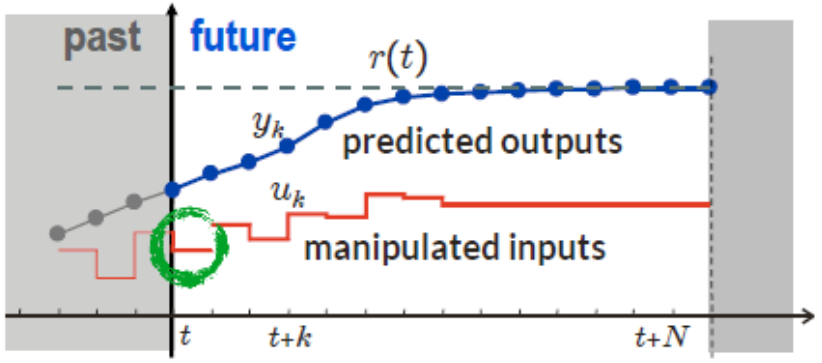
- $z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} \in \mathbb{R}^{Nm}$ is the optimization vector



- $H = H' \succ 0$, and H, F, Y, G, W, S depend on weights Q, R, P upper and lower bounds $u_{\min}, u_{\max}, y_{\min}, y_{\max}$ and model matrices A, B, C.

- **线性MPC——有约束问题(状态镇定)**

@ each sampling step t:



- Measure (or estimate) the current state $x(t)$
- Get the solution $z^* = \begin{bmatrix} u_0^* \\ u_1^* \\ \vdots \\ u_{N-1}^* \end{bmatrix}$ of the QP

$$\begin{cases} \min\limits_{z} & \frac{1}{2} z' H z + \overbrace{x'(t) F' z}^{\text{feedback}} \\ \text{s.t.} & Gz \leq W + S \underbrace{x(t)}_{\text{feedback}} \end{cases}$$

- Apply only $u(t) = u_0^*$, discarding the remaining optimal inputs $u_1^*, \ldots, u_{N-1}^*$

- **线性MPC——有约束问题(输出跟踪)**

  - Objective: make the output $y(t)$ track a reference signal $r(t)$
  - Let us parameterize the problem using the **input increments**

$$\Delta u(t) = u(t) - u(t-1)$$

- LQR问题的MPC提法
- **输出跟踪问题**

  - As $u(t) = u(t-1) + \Delta u(t)$ we need to extend the system with a new state $x_u(t) = u(t-1)$

$$\begin{cases} x(t+1) &= Ax(t) + Bu(t-1) + B\Delta u(t) \\ x_u(t+1) &= x_u(t) + \Delta u(t) \end{cases}$$

$$\begin{cases} \begin{bmatrix} x(t+1) \\ x_u(t+1) \end{bmatrix} &= \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x(t) \\ x_u(t) \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u(t) \\ y(t) &= \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ x_u(t) \end{bmatrix} \end{cases}$$

  - Again a linear system with states $x(t), x_u(t)$ and input $\Delta u(t)$

- **线性MPC——有约束问题(输出跟踪)**

➢ MPC问题的推导

- Optimal control problem (quadratic performance index):

$$\begin{aligned}
\min_{z} \quad & \sum_{k=0}^{N-1} \|W^{y}(y_{k+1} - r(t))\|_2^2 + \|W^{\Delta u}\Delta u_k\|_2^2 \\
& [\Delta u_k \triangleq u_k - u_{k-1}],\ u_{-1} = u(t-1) \\
\text{s.t.} \quad & u_{\min} \le u_k \le u_{\max},\ k = 0, \dots, N-1 \\
& y_{\min} \le y_k \le y_{\max},\ k = 1, \dots, N \\
& \Delta u_{\min} \le \Delta u_k \le \Delta u_{\max},\ k = 0, \dots, N-1
\end{aligned}$$

$$z = \begin{bmatrix} \Delta u_0 \\ \Delta u_1 \\ \vdots \\ \Delta u_{N-1} \end{bmatrix} \text{ or } z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

weight $W^{\cdot}$ = diagonal matrix (more generally, Cholesky factor of $Q^{\cdot} = (W^{\cdot})'W^{\cdot}$)

$$\begin{aligned}
\min_{z} \quad & J(z, x(t)) = \tfrac{1}{2} z'Hz + [x'(t)\, r'(t)\, u'(t-1)]F'z \\
\text{s.t.} \quad & Gz \le W + S \begin{bmatrix} x(t) \\ r(t) \\ u(t-1) \end{bmatrix}
\end{aligned}$$

convex Quadratic Program

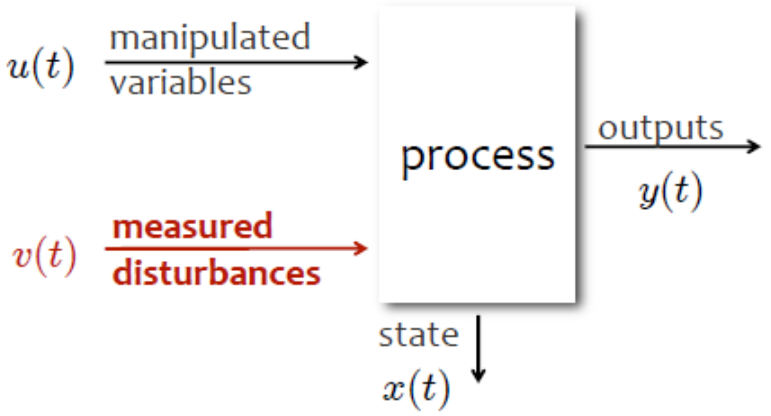
- Add the extra penalty $\|W^{u}(u_k - u_{\text{ref}}(t))\|_2^2$ to track input references
- Constraints may depend on $r(t)$, such as $e_{\min} \le y_k - r(t) \le e_{\max}$

- **线性MPC——有约束问题(模型误差)**

➢ MPC问题的推导

- Measured disturbance $v(t)$ = input that is measured but not manipulated

$$\begin{cases} x_{k+1} &= Ax_k + Bu_k + B_v v(t) \\ y_k &= Cx_k + D_v v(t) \end{cases}$$



$$x_k = A^k x_0 + \sum_{j=0}^{k-1} A^j B u_{k-1-j} + A^j B_v v(t)$$

- Same performance index, same constraints. We still have a QP:

$$\min_z \quad \tfrac{1}{2} z' H z + [x'(t)\, r'(t)\, u'(t-1)\, v'(t)] F' z$$

$$\text{s.t.} \quad Gz \le W + S \begin{bmatrix} x(t) \\ r(t) \\ u(t-1) \\ v(t) \end{bmatrix}$$

- Note that MPC naturally provides **feedforward action** on $v(t)$ and $r(t)$

- **MPC & LQR**

  - Special case: $J(z, x_0) = x'_N P x_N + \sum_{k=0}^{N-1} x'_k Q x_k + u'_k R u_k, N_u = N$, with matrix P solving the Algebraic Riccati Equation

➢ 无约束MPC与LQR

$$P = A'PA - A'PB(B'PB + R)^{-1}B'PA + Q$$

Jacopo Francesco Riccati
(1676–1754)

(unconstrained) MPC = LQR for any choice of the prediction horizon N

Proof: : Easily follows from Bellman's principle of optimality (dynamic programming): $x'_N P x_N$ = optimal "cost-to-go" from time N to ∞.

- **MPC & LQR**

➢ 有约束MPC与LQR

- Consider again the constrained MPC law based on minimizing

$$\begin{aligned} \min_{z} \quad & x_N' P x_N + \sum_{k=0}^{N-1} x_k' Q x_k + u_k' R u_k \\ \text{s.t.} \quad & u_{\min} \le u_k \le u_{\max},\ k = 0, \dots, N-1 \\ & y_{\min} \le y_k \le y_{\max},\ k = 1, \dots, N \\ & u_k = K x_k,\ k = N_u, \dots, N-1 \end{aligned}$$

- Choose matrix P and terminal gain K by solving the LQR problem

$$\begin{aligned} K &= -(R + B'PB)^{-1} B'PA \\ P &= (A + BK)'P(A + BK) + K'RK + Q \end{aligned}$$

- In a polyhedral region around the origin, **constrained MPC = constrained LQR** for any choice of the prediction and control horizons N, N_u

(Sznaier, Damborg, 1987) (Chmielewski, Manousiouthakis, 1996) (Scokaert, Rawlings, 1998) (Bemporad, Morari, Dua Pistikopoulos, 2002)

- The larger the horizon N, the larger the region where MPC $\equiv$ constrained LQR

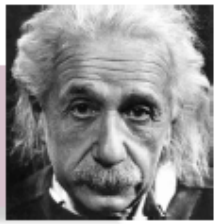
- **简化模型**

- Computation complexity depends on chosen prediction model
- Good models for MPC must be
  - **Descriptive** enough to capture the most significant dynamics of the system

大道至简

  - **Simple** enough for solving the optimization problem

"Things should be made as simple as possible, but not any simpler."



Albert Einstein
(1879–1955)

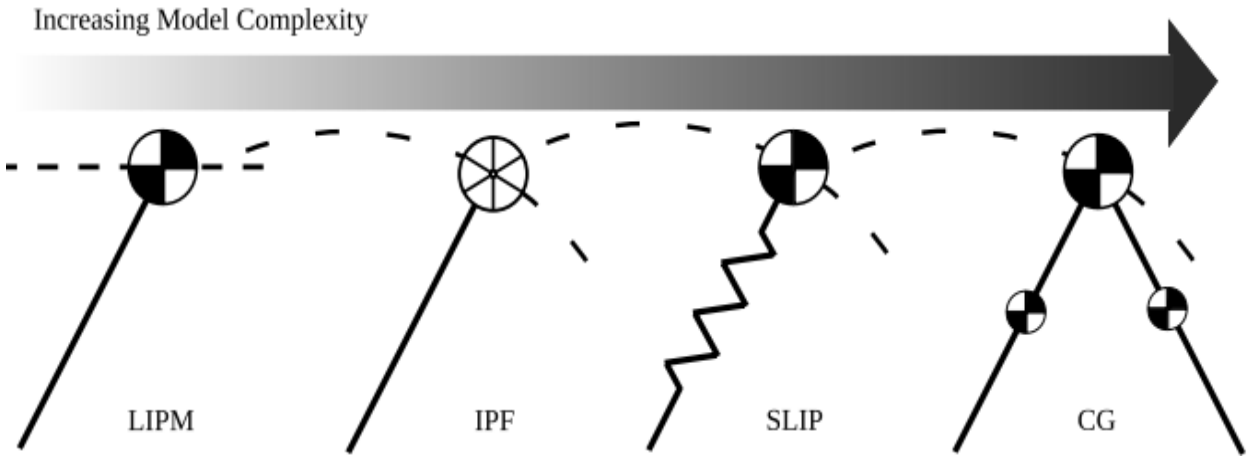
- **平衡矛盾**

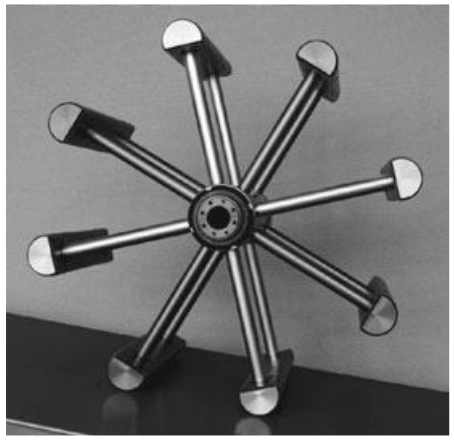
➢ 尽可能充分的描述性来捕捉系统中最重要的动态(模型描述的复杂性)

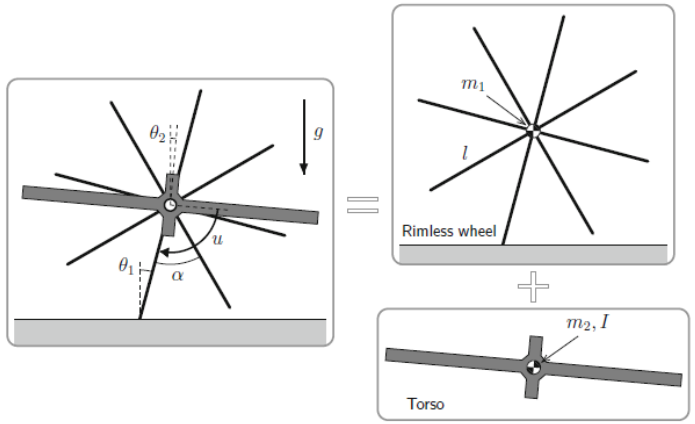
➢ 对于解决优化问题来说尽可能简单(模型计算的简单性)

- **模型简化**

  - 轮式 & 多轴
  - 多轴 & 两轴
  - 刚性 & 弹性
  - 单杆 & 多杆







Prahlad Vadakkepat. Humanoid Robotics: A Reference[J]. 2017.

- **模型简化特点**

  - 轨迹优化通常需要基于动力学模型，只有利用机器人动力学特性才能产生优雅的运动
  - 实时控制系统无法支撑复杂模型的计算量，通常需要做模型简化
  - 高效的动力学模型：用简化模型去描述复杂系统的动态特性

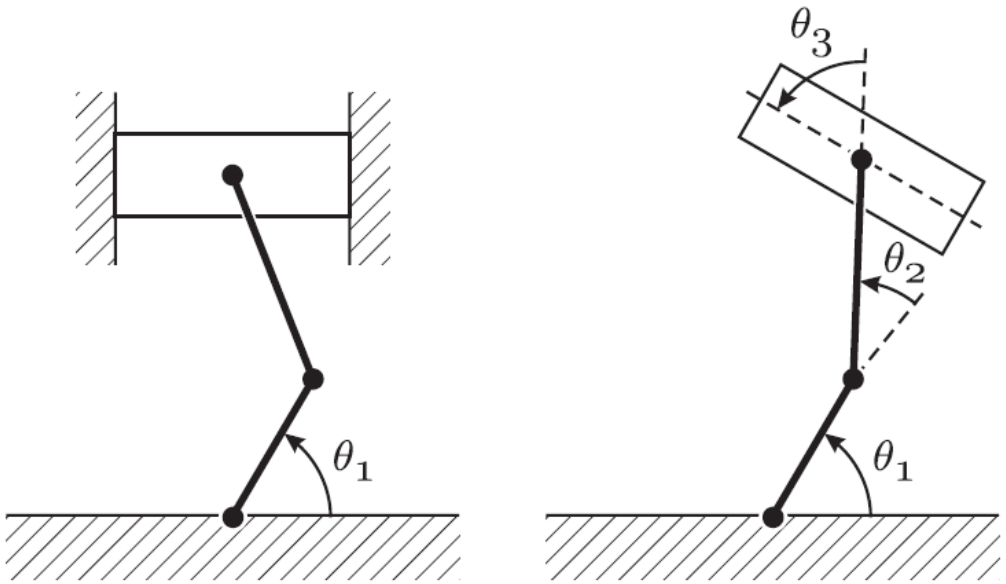
  - 模型思路：上层轨迹优化基于简化模型，下层轨迹跟踪基于复杂模型
  - 底层期望轨迹符合实际机器人动力学特性，同时能够跟踪上层轨迹

12 简化模型与内动态

- **虚约束**

  - (a) 受气缸壁约束的单自由度平面活塞系统
  - (b) 不受约束的三自由度平面活塞系统

  - 受物理约束的(a)系统 = 自由系统(b) + 虚约束



(a) (b)

虚约束的实现方法:

- 通过反馈控制渐近地在动态系统上施加完整约束

$$0 = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2),$$
$$\pi = \theta_1 + \theta_2 + \theta_3,$$

虚约束

$$y_1 = L_1 \cos(\theta_1) + L_2 \cos(\theta_1 + \theta_2),$$
$$y_2 = \theta_1 + \theta_2 + \theta_3 - \pi.$$
$$y_1 = \theta_2 - \left(\pi - \theta_1 - \arccos\left(\frac{L_1}{L_2}\cos(\theta_1)\right)\right),$$
$$y_2 = \theta_3 - \arccos\left(\frac{L_1}{L_2}\cos(\theta_1)\right).$$

反馈系统

12 简化模型与内动态

- **内动态**
  - 非线性系统的反馈线性化
  - 微分同胚，李导数与李括号，相对阶

输入输出标准型

假定单输入单输出非线性系统在 x_0 的相对阶为 r，令

$$z_1=\phi_1(x)=h(x)$$
$$z_2=\phi_2(x)=L_fh(x)$$
$$\vdots$$
$$z_r=\phi_r(x)=L_f^{r-1}h(x)$$

特点：

1. $\nabla h,\nabla L_fh,\cdots,\nabla L_f^{r-1}h$ 线性无关
2. $L_gh=0,\quad L_gL_fh=0,\cdots,L_gL_f^{r-2}h=0,\quad L_gL_f^{r-2}h\neq 0$

$\nabla h,\nabla L_fh,\cdots,\nabla L_f^{r-2}h$ 在与g正交的空间里

若r<n

总能找到 n-r 个非线性函数

$$z_{r+1}=\phi_{r+1}(x)$$
$$\vdots$$
$$z_n=\phi_n(x)$$

使得非线性变换 $z=\phi(x)$ 的雅可比矩阵在 x^0 非奇异

$$z=\phi(x)=\begin{bmatrix}\phi_1(x)\\ \phi_2(x)\\ \vdots\\ \phi_n(x)\end{bmatrix}$$

特别地

$$L_g\phi_i(x)=0,\quad r+1\le i\le n$$

$\Rightarrow$

$\dot{z}_i(r+1\le i\le n)$ 不显式地依赖于输入u

$$\dot{z}_1=z_2$$
$$\dot{z}_2=z_3$$
$$\vdots$$
$$\dot{z}_{r-1}=z_r$$
$$\dot{z}_r=a(z)+b(z)u$$
$$\dot{z}_{r+1}=L_f\phi_{r+1}(x)=L_f\phi_{r+1}(\phi^{-1}(z))=q_{r+1}(z)$$
$$\vdots$$
$$\dot{z}_n=L_f\phi_n(x)=L_f\phi_n(\phi^{-1}(z))=q_n(z)$$

其中 $a(z)=L_f^rh,\quad b(z)=L_gL_f^{r-1}h$

JJE Slotine, WP Li. Applied Nonlinear Control. 1991.

- **内动态**

  - 标准型
  - 内动态
  - 零动态与最小相位
  - 局部镇定问题

状态方程变成

$$\dot{\xi}=\begin{bmatrix}\xi_2\\ \vdots\\ \xi_r\\ a(\xi,\eta)\end{bmatrix}+\begin{bmatrix}0\\ \vdots\\ 0\\ b(\xi,\eta)\end{bmatrix}u$$

$$\dot{\eta}=q(\xi,\eta)$$

$$y=\begin{bmatrix}1 & 0 & \cdots & 0\end{bmatrix}\cdot\xi$$

标准型

途径：对非线性系统定义一个零动态子系统。

零动态子系统定义为当系统的输出被输入强制为零时的内动态子系统

$\mu=0$，系统的内动态方程可以写成

$$\dot{\psi}=\omega(0,\psi)$$

零动态子系统

令

$$v=-k_{r-1}y^{(r-1)}-\cdots-k_1\dot{y}-k_0y \qquad (1)$$

式中系数k_i的选择应使多项式

$$K(p)=p^r+k_{r-1}p^{r-1}+\cdots+k_1p+k_0 \qquad (2)$$

的根全部位于左半平面。

实际输入 u 为

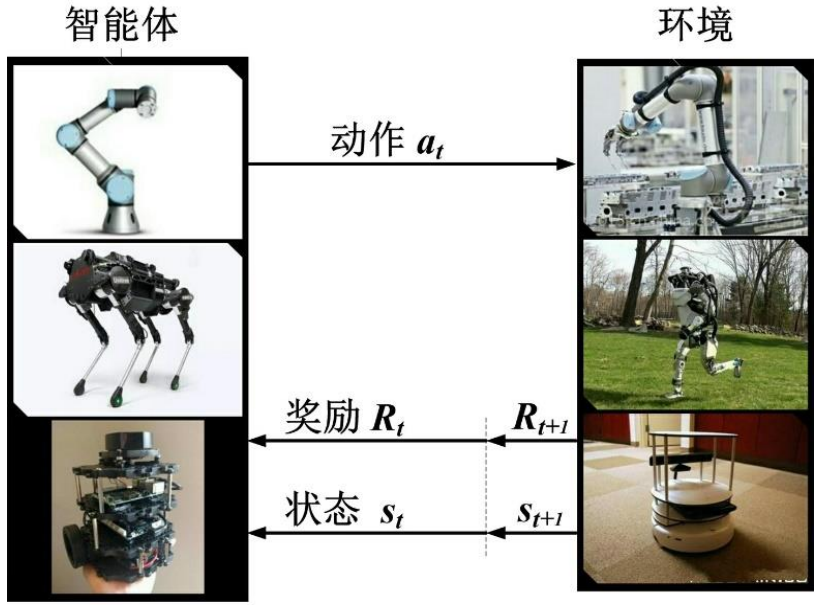
$$u(x)=\frac{1}{L_gL_f^{r-1}y}\left[-L_f^ry-k_{r-1}y^{(r-1)}-\cdots-k_1\dot{y}-k_0y\right]$$

JJE Slotine, WP Li. Applied Nonlinear Control. 1991.

- **动力学模型**
  - 确定性模型：动作执行、状态感知都是确定的
  - 不确定模型：动力学模型，状态观测均为不确定的

    - 处理方案
      - 方式1：非概率模型
        - 动力学模型：系统辨识与自适应控制(参数估计)
        - 状态观测：Kalman滤波等(状态估计)
      - 方法2：概率模型
        - 动力学模型：马尔科夫决策过程(MDP)
        - 状态观测：马尔科夫决策过程(MDP)

- **马尔科夫决策过程**



- 机器人任务可以形式化为一个MDP过程
- 智能体在状态$\boldsymbol{s}_t$处采取动作$\boldsymbol{a}_t$，获得即时奖励R_{t+1}，并通过动力学$P(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t,\boldsymbol{a}_t)$转移到状态$\boldsymbol{s}_{t+1}$
- 设置一个完全自主的智能体，通过反复试验与环境进行交互，进而学习最优行为

交互轨迹：$\tau=(\boldsymbol{s}_0,\boldsymbol{a}_0,R_1,\boldsymbol{s}_1,\boldsymbol{a}_1,R_2\ldots,\boldsymbol{s}_T)$

值函数：$Q^{\pi}(\boldsymbol{s},\boldsymbol{a})=\mathbb{E}_{\pi}\left[\sum_{k=t}^{T}\gamma^{k-t}R_{k+1}|\boldsymbol{s}_t=\boldsymbol{s},\boldsymbol{a}_t=\boldsymbol{a}\right]$

策略：$\pi(\boldsymbol{a}|\boldsymbol{s})$：随机性策略，动作的选取服从随机概率分布，$\boldsymbol{a}\sim\pi(\boldsymbol{a}|\boldsymbol{s})$

$\mu(\boldsymbol{s})$：确定性策略，动作的选择满足映射关系，$\boldsymbol{a}=\mu(\boldsymbol{s})$

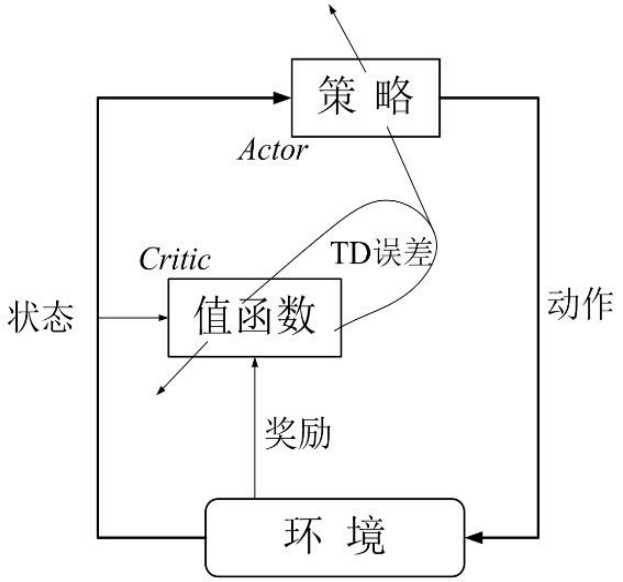
累积回报：$G_t=R_{t+1}+\gamma R_{t+2}+\gamma^2R_{t+3}+\cdots+\gamma^{T-t}R_{T+1}=\sum_{k=t}^{T}\gamma^{k-t}R_{k+1}$

强化学习目标：$J=\mathbb{E}_{\boldsymbol{s}\sim\rho,\boldsymbol{a}\sim\pi}[G_1]$ 通过寻找最优策略，最大化累积回报的期望

- **强化学习算法**

基于行动者评论家的方法（Actor-Critic）

- 结合基于值函数和基于策略两种方法的优势
- Critic网络采用TD误差学习算法实现值函数的估计
- Actor网络则利用策略梯度方法进行梯度下降学习



基于值函数的方法（Critic）
1. TD学习

$$Q^{\pi}(\boldsymbol{s}_t, \boldsymbol{a}_t) \leftarrow Q^{\pi}(\boldsymbol{s}_t, \boldsymbol{a}_t) + \alpha\delta_t$$
$$\delta_t = y_t - Q^{\pi}(\boldsymbol{s}_t, \boldsymbol{a}_t)$$

Q学习： $y_t = R_{t+1} + \gamma \max_{\boldsymbol{a}'} Q^{\pi}(\boldsymbol{s}_{t+1}, \boldsymbol{a}')$
SARSA： $y_t = R_{t+1} + \gamma Q^{\pi}(\boldsymbol{s}_{t+1}, \boldsymbol{a}_{t+1})$
2. Monte-Carlo方法

基于策略的方法（Actor）
1. 无梯度的方法
2. 基于策略梯度的方法

$$\mathcal{J}(\pi_\theta) = \mathbb{E}_{\pi_\theta}\left[f_{\pi_\theta}(\cdot)\right]$$

随机： $\nabla_\theta \mathcal{J}(\pi_\theta) = \mathbb{E}_{\boldsymbol{s},\boldsymbol{a}}[\nabla_\theta \log \pi_\theta(\boldsymbol{s}|\boldsymbol{a}) \cdot Q^{\pi_\theta}(\boldsymbol{s}, \boldsymbol{a})]$
确定： $\nabla_\theta \mathcal{J}(\pi_\theta) = \mathbb{E}_{\boldsymbol{s}}\left[\nabla_\theta \mu_\theta(\boldsymbol{s}) \cdot \nabla_{\boldsymbol{a}} Q^{\omega}(\boldsymbol{s}, \boldsymbol{a})|_{\boldsymbol{a}=\mu_\theta(\boldsymbol{s})}\right]$

- **广义机器人动力学**

无模型的强化学习

- 假设不知道初始分布$P(\boldsymbol{s}_0)$和状态转移概率$P(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$
- 优点：可以不用对环境建模，或是快速将强化学习算法应用于不熟悉的领域
- 缺点：智能体只能探索环境，数据利用率低，对于大型网络很难产生好的数据

基于模型的强化学习

- 模型描述了关于环境行为和智能体对环境影响的基本预测信息
- 狭义动力学：传统机器人依赖于基于人类对物理的洞察而手工生成的模型
- 广义动力学：自主机器人能够从可访问的数据流中提取信息，自动生成模型

- 基于模型的方法
  基于任务建模，通用的方法不一定有效
- 学习动力学模型的作用
  - ✓ 退化为动态规划与最优控制的问题
  - ✓ 进一步训练，稳定网络
  - ✓ 进行示范学习

- **强化学习拓展(深度学习)**

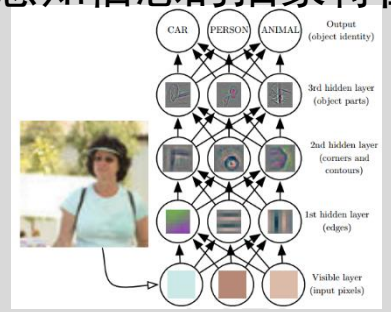
强化学习的问题：

- 机器学习算法固有的问题：内存复杂度与计算复杂度，强化学习存在维度灾难
- 自主智能机器人两个基本要素：感知和控制，而强化学习无法解决感知问题

深度学习

1. 表示学习

从原始高维数据中提取有效感知信息的抽象特征



2. 函数逼近

3层神经网络就可以逼近任意非线性函数

实现了对连续状态和动作空间的值函数和策略的存储和表达

3. 分层分布式

使得中间过程可以拓展为并行化计算

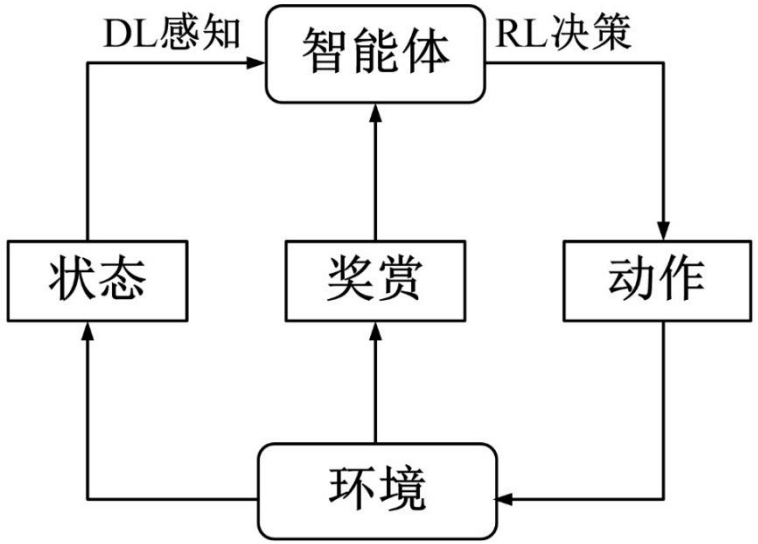
多线程、GPU加速、多机框架

- **深度强化学习**

深度强化学习（DRL）

➢ 结合深度学习的感知能力和强化学习的决策能力

➢ 实现了从原始输入到输出的端对端直接控制



深度强化学习基本过程

（1）智能体与环境进行交互并得到高维原始状态信息，通过深度学习来进行处理，得到低维抽象的特征表示

（2）基于期望回报来评价动作值函数，并通过策略网络来生成动作

（3）智能体基于动作与环境交互得到下一个观测值

（4）重复（1）~（3），直至得到最优策略

深度强化学习算法分类

➢ 状态空间和动作空间的维度

➢ 是否依赖动力学模型

➢ 奖励函数是否已知

- **概率图模型**
  - 贝叶斯网络
  - 马尔科夫随机场

待完善