- Formulation of nonlinear systém equations
- Linearization of continuous nonlinear model
- Diskretization of linearized state-space model
- Evaluation of control inputs based on the system motion request and system dynamics prediction

The basic form of dynamical equations:

The typical form of dynamic equations of the mechanical part of robots comes from the Lagrange equations of the mixed type numerically transformed to the independent coordinates.

$$R^T M R \ddot{q} + R^T M \dot{R} \dot{q} = R^T g + R^T T u$$

These equations can be extended by the models of electrical drives, compliancies and other dynamical properties. The typical advantage of the predictive control of robots and other mechanisms is the availability of physical models which can be procesed (for example linearized) exactly.

Discretization:

The relationships between the discrete state-space matrices **A, B, C, D** and the continuous-time state-space matrices **F, G, H, D** are given for piece-wise-constant input, as follows:

$$A = e^{FT}$$
$$B = \int_0^T e^{F\tau} G d\tau$$
$$C = H$$

Basic algorithm of predictive control:

The local behaviour can be described as a linear model by a classical discrete state description with locally constant state matrixes **A**, **B** and **C**. The matrix **D** is considered zero.

$$\Delta\mathbf{x}_{k+1} = \mathbf{A}\Delta\mathbf{x}_k + \mathbf{B}\mathbf{u}_k$$
$$\mathbf{y}_k = \mathbf{C}\Delta\mathbf{x}_k \qquad (1)$$

where **Δx** are system states and **y** are investigated outputs. The transfer function is then transformed into the state description. The predictive control can be applied to the model created using the above described procedure.

Writing the relation for N subsequent steps as follows,

$$\begin{aligned}
\mathbf{y}_k &= \mathbf{C}\Delta\mathbf{x}_k \\
\Delta\hat{\mathbf{x}}_{k+1} &= \mathbf{A}\Delta\mathbf{x}_k + \mathbf{B}\mathbf{u}_k \\
\hat{\mathbf{y}}_{k+1} &= \mathbf{C}\mathbf{A}\Delta\mathbf{x}_k + \mathbf{C}\mathbf{B}\mathbf{u}_k \\
&\vdots \\
\Delta\hat{\mathbf{x}}_{k+N} &= \mathbf{A}^N\Delta\mathbf{x}_k + \mathbf{A}^{N-1}\mathbf{B}\mathbf{u}_k + \ldots + \mathbf{B}\mathbf{u}_{k+N+1} \\
\hat{\mathbf{y}}_{k+N} &= \mathbf{C}\mathbf{A}^N\Delta\mathbf{x}_k + \mathbf{C}\mathbf{A}^{N-1}\mathbf{B}\mathbf{u}_k + \ldots + \mathbf{C}\mathbf{B}\mathbf{u}_{k+N+1}
\end{aligned} \qquad (2)$$

and rewriting these equations into the matrix form, we obtain the prediction of the outputs

$$\hat{\boldsymbol{y}} = \boldsymbol{f} + \boldsymbol{G}\boldsymbol{u} \qquad (3)$$

where

$$\mathbf{f} = \begin{bmatrix} \mathbf{CA} \\ \vdots \\ \mathbf{CA}^N \end{bmatrix} \mathbf{x}_k \quad , \quad \mathbf{G} = \begin{bmatrix} \mathbf{CB} & \dots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{CA}^{N-1}\mathbf{B} & \dots & \mathbf{CB} \end{bmatrix} \quad (4)$$

The control is derived from the optimization of a quadratic performance index. The performance index Jk is optimized in the step k using the prediction $\hat{\mathbf{y}} = [\hat{\mathbf{y}}_{k+1} \quad \cdots \quad \hat{\mathbf{y}}_{k+N}]^T$ for **y**.

$$\begin{aligned} J_k &= \boldsymbol{\varepsilon}\left\{(\hat{\boldsymbol{y}} - \boldsymbol{w})^T \boldsymbol{Q}(\hat{\boldsymbol{y}} - \boldsymbol{w}) + \boldsymbol{u}^T \boldsymbol{p}\boldsymbol{u}\right\} = \\ &= \boldsymbol{\varepsilon}\left\{(\boldsymbol{G}\boldsymbol{u} + \boldsymbol{f} - \boldsymbol{w})^T \boldsymbol{Q}(\boldsymbol{G}\boldsymbol{u} + \boldsymbol{f} - \boldsymbol{w}) + \boldsymbol{u}^T \boldsymbol{p}\boldsymbol{u}\right\} \end{aligned} \quad (5)$$

where ε is a mean value operator, N is the prediction horizon, **y** is the output vector, **w** is the desired output vector (for robot typically the requested trajectory), **Q** is a penalization matrix of the outputs, **p** is a penalization of the inputs,

and $\boldsymbol{u} = [\boldsymbol{u}_k, \ \dots, \boldsymbol{u}_{k+N+1}]^{\mathrm{T}}$ is the input vector. From the requirement of the minimization of the performance index

$$J_k = \min_{\boldsymbol{u}} J_k(\boldsymbol{u}) \quad (6)$$

the control law is derived

$$\boldsymbol{u} = (\boldsymbol{G}^T \boldsymbol{Q}\boldsymbol{G} + \boldsymbol{p})^{-1} \boldsymbol{G}^T \boldsymbol{Q}(\boldsymbol{w} - \boldsymbol{f}) \quad (7)$$

However, only the first element of the vector ***u*** is used for the nearest control action.

The control problem very often includes the constraints of the values of different variables. In case of the robots, the constrained values may involve operation constraints (e.g. drive limits, problems of colisions with environment etc.) In order to incorporate these constraints, the direct computation procedure (7) must be generalized. Quadratic programming appears to be a good tool for this purpose. The unconstrained optimization problem (6) is reformulated as a quadratic optimization problem

$$\min_{\boldsymbol{u}} \left(\frac{1}{2} \boldsymbol{u}^T \boldsymbol{H}_{\mathrm{QP}} \boldsymbol{u} + \boldsymbol{f}_{\mathrm{QP}}^{\mathrm{T}} \boldsymbol{u} \right) \quad (8)$$

with a constraint condition

$$\boldsymbol{A}_{\mathrm{QP}} \boldsymbol{u} \le \boldsymbol{b}_{\mathrm{QP}} \quad (9)$$

The particular matrices used for quadratic programming follow from equations (3)-(7)

$$\begin{aligned} \mathbf{H}_{\mathrm{QP}} &= \mathbf{G}^T \mathbf{Q} \mathbf{G} + \mathbf{p} \\ \mathbf{f}_{\mathrm{QP}} &= \mathbf{G}^T \mathbf{Q}(\mathbf{w} - \mathbf{f}) \end{aligned} \quad (10)$$

This optimization problem is solved in every sampling instant. Again, only the first element of vector ***u*** is used for the nearest control action.