Optimal Input Signal Design & MPC of Nonlinear Dynamical Systems

Mariette Annergren

Supervisors: Bo Wahlberg, KTH, and Yang Wang, Stanford University.

August 27, 2010

Outline

Introduction

- MPC stands for *Model Predictive Control*.

Introduction

- MPC stands for *Model Predictive Control*.
- It is a controller based on predictions of states and minimization of a cost function.

Introduction

- MPC stands for *Model Predictive Control*.
- It is a controller based on predictions of states and minimization of a cost function.
- Our objective is to create a general method of implementing MPC of *nonlinear* dynamical systems.

Introduction

- MPC stands for *Model Predictive Control*.
- It is a controller based on predictions of states and minimization of a cost function.
- Our objective is to create a general method of implementing MPC of *nonlinear* dynamical systems.
- It is problematic to use MPC on *nonlinear* dynamical systems.

Introduction (cont)

It is still interesting to use MPC on nonlinear dynamical systems.

Introduction (cont)

It is still interesting to use MPC on nonlinear dynamical systems.

- We can add explicit constraints on states and input signals.

Introduction (cont)

It is still interesting to use MPC on nonlinear dynamical systems.

- We can add explicit constraints on states and input signals.
- MPC does not require a lot of tuning of its parameters.

Introduction (cont)

Some existing methods of nonlinear MPC are based on

Introduction (cont)

Some existing methods of nonlinear MPC are based on

- making the problem convex through a change of variables, (only possible in some special cases)

Introduction (cont)

Some existing methods of nonlinear MPC are based on

- making the problem convex through a change of variables, (only possible in some special cases)
- nonlinear programming. (difficult, approximative solution)

Introduction (cont)

Some existing methods of nonlinear MPC are based on

- making the problem convex through a change of variables, (only possible in some special cases)
- nonlinear programming. (difficult, approximative solution)

In our method we transform the optimization problem to a convex problem by *linearizing the nonlinear system dynamics*.

The MPC

The MPC algorithm is based on

The MPC

The MPC algorithm is based on

- measured data from the system,

The MPC

The MPC algorithm is based on

- measured data from the system,
- a mathematical model of the system,

The MPC

The MPC algorithm is based on

- measured data from the system,
- a mathematical model of the system,
- a cost function which penalizes undesirable behavior.

The steps of the MPC Algorithm

The steps of MPC can be summarized as

The steps of the MPC Algorithm

The steps of MPC can be summarized as

- get data from the system,

The steps of the MPC Algorithm

The steps of MPC can be summarized as

- get data from the system,
- predict future output signals,

The steps of the MPC Algorithm

The steps of MPC can be summarized as

- get data from the system,
- predict future output signals,
- insert the predictions in the cost function,

The steps of the MPC Algorithm

The steps of MPC can be summarized as

- get data from the system,
- predict future output signals,
- insert the predictions in the cost function,
- minimize it with respect to a future input signal sequence,

The steps of the MPC Algorithm

The steps of MPC can be summarized as

- get data from the system,
- predict future output signals,
- insert the predictions in the cost function,
- minimize it with respect to a future input signal sequence,
- apply the first input signal in the obtained optimal input signal sequence,

The steps of the MPC Algorithm

The steps of MPC can be summarized as

- get data from the system,
- predict future output signals,
- insert the predictions in the cost function,
- minimize it with respect to a future input signal sequence,
- apply the first input signal in the obtained optimal input signal sequence,
- repeat procedure from the first step.

The MPC (cont)

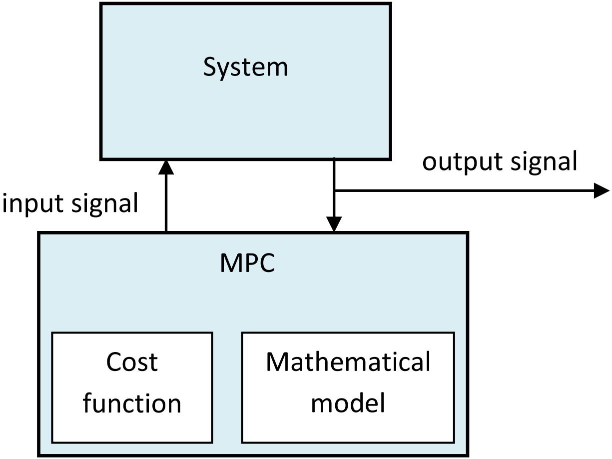

Figure: MPC

Requirements of the Implementation

There are basically three parts necessary to implement an MPC according to our procedure:

Requirements of the Implementation

There are basically three parts necessary to implement an MPC according to our procedure:

- Simulation.

Requirements of the Implementation

There are basically three parts necessary to implement an MPC according to our procedure:

- Simulation.
- Linearization/Discretization.

Requirements of the Implementation

There are basically three parts necessary to implement an MPC according to our procedure:

- Simulation.
- Linearization/Discretization.
- MPC algorithm.

The Procedure

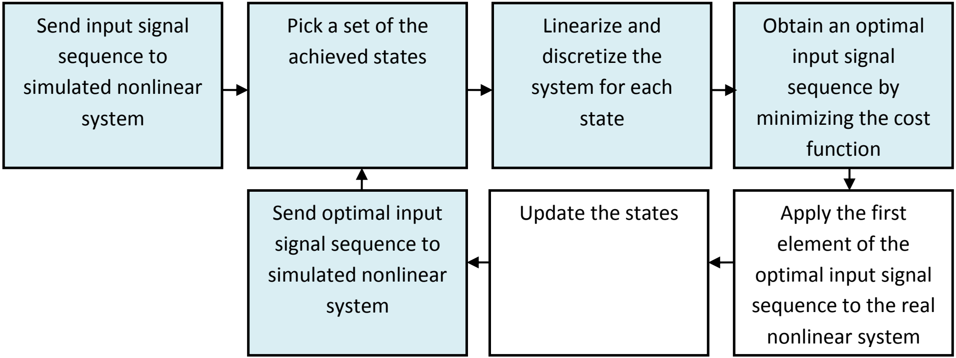

Figure: Procedure

Two Link Robot Arm

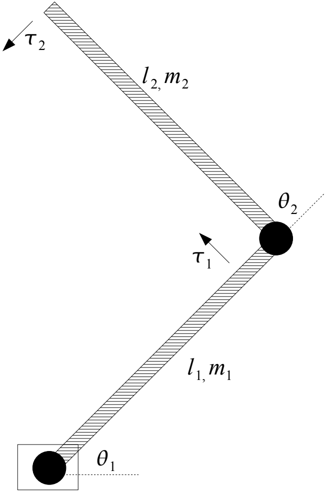

Figure: Two Link Robot Arm

Nonlinear System Description

$$M(\Theta(t))\ddot{\Theta}(t) + W(\Theta(t), \dot{\Theta}(t))\dot{\Theta}(t) = \tau(t),$$

Nonlinear System Description

$$M(\Theta(t))\ddot{\Theta}(t) + W(\Theta(t), \dot{\Theta}(t))\dot{\Theta}(t) = \tau(t),$$

with

$$M(\Theta(t)) = \begin{pmatrix} (m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \mathrm{c}_2(t) & m_2 l_2^2 + m_2 l_1 l_2 \mathrm{c}_2(t) \\ m_2 l_2^2 + m_2 l_1 l_2 \mathrm{c}_2(t) & m_2 l_2^2 \end{pmatrix}$$

Nonlinear System Description

$$M(\Theta(t))\ddot{\Theta}(t) + W(\Theta(t), \dot{\Theta}(t))\dot{\Theta}(t) = \tau(t),$$

with

$$M(\Theta(t)) = \begin{pmatrix} (m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \mathrm{c}_2(t) & m_2 l_2^2 + m_2 l_1 l_2 \mathrm{c}_2(t) \\ m_2 l_2^2 + m_2 l_1 l_2 \mathrm{c}_2(t) & m_2 l_2^2 \end{pmatrix}$$

and

$$W(\Theta(t), \dot{\Theta}(t)) = \begin{pmatrix} -2m_2 l_1 l_2 \dot{\theta}_2(t)\mathrm{s}_2(t) & -m_2 l_1 l_2 \dot{\theta}_2(t)\mathrm{s}_2(t) \\ -m_2 l_1 l_2 \dot{\theta}_1(t)\mathrm{s}_2(t) & 0 \end{pmatrix},$$

Nonlinear System Description

$$M(\Theta(t))\ddot{\Theta}(t) + W(\Theta(t),\dot{\Theta}(t))\dot{\Theta}(t) = \tau(t),$$

with

$$M(\Theta(t)) = \begin{pmatrix} (m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \mathrm{c}_2(t) & m_2 l_2^2 + m_2 l_1 l_2 \mathrm{c}_2(t) \\ m_2 l_2^2 + m_2 l_1 l_2 \mathrm{c}_2(t) & m_2 l_2^2 \end{pmatrix}$$

and

$$W(\Theta(t),\dot{\Theta}(t)) = \begin{pmatrix} -2m_2 l_1 l_2 \dot{\theta}_2(t)\mathrm{s}_2(t) & -m_2 l_1 l_2 \dot{\theta}_2(t)\mathrm{s}_2(t) \\ -m_2 l_1 l_2 \dot{\theta}_1(t)\mathrm{s}_2(t) & 0 \end{pmatrix},$$

where $\mathrm{s}_i(t) = \sin(\theta_i(t))$ and $\mathrm{c}_i(t) = \cos(\theta_i(t))$, for $i = 1, 2$.

Nonlinear System Description

$$M(\Theta(t))\ddot{\Theta}(t) + W(\Theta(t), \dot{\Theta}(t))\dot{\Theta}(t) = \tau(t),$$

with

$$M(\Theta(t)) = \begin{pmatrix} (m_1 + m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \mathrm{c}_2(t) & m_2 l_2^2 + m_2 l_1 l_2 \mathrm{c}_2(t) \\ m_2 l_2^2 + m_2 l_1 l_2 \mathrm{c}_2(t) & m_2 l_2^2 \end{pmatrix}$$

and

$$W(\Theta(t), \dot{\Theta}(t)) = \begin{pmatrix} -2m_2 l_1 l_2 \dot{\theta}_2(t)\mathrm{s}_2(t) & -m_2 l_1 l_2 \dot{\theta}_2(t)\mathrm{s}_2(t) \\ -m_2 l_1 l_2 \dot{\theta}_1(t)\mathrm{s}_2(t) & 0 \end{pmatrix},$$

where $\mathrm{s}_i(t) = \sin(\theta_i(t))$ and $\mathrm{c}_i(t) = \cos(\theta_i(t))$, for $i = 1, 2$.

Highly nonlinear and quite complicated!

Simulation

The well-known Runge-Kutta method of order four is used to discretize and simulate the nonlinear system.

Linearization/Discretization

The linearization and discretization occur simultaneously.

Linearization/Discretization

The linearization and discretization occur simultaneously.

$$\begin{aligned}\tau_k &= M(\Theta_k^{disc})(\dot{\Theta}_{k+h} - \dot{\Theta}_k)\frac{1}{h} + \\ &\quad + W(\Theta_k^{disc}, \Theta_k^{disc})(\Theta_{k+h} - \Theta_k)\frac{1}{h},\end{aligned}$$

with

$$\dot{\Theta}_k = (\Theta_{k+h} - \Theta_k)\frac{1}{h}.$$

MPC Algorithm: Objective Function

The objective is to make the end point of the robot arm follow a pre-specified trajectory, detonated p^F.

MPC Algorithm: Objective Function

The objective is to make the end point of the robot arm follow a pre-specified trajectory, detonated p^F.

The objective function is

$$f_o(x) = \sum_{k=0}^{T} \left\| \begin{bmatrix} l_1 \mathrm{c}(\hat{\theta}_{1,k}) + l_2 \mathrm{c}(\hat{\theta}_{1,k} + \hat{\theta}_{2,k}) \\ l_1 \mathrm{s}(\hat{\theta}_{1,k}) + l_2 \mathrm{s}(\hat{\theta}_{1,k} + \hat{\theta}_{2,k}) \end{bmatrix} - p_k^F \right\|_2^2,$$

where $\mathrm{s}(\cdot) = \sin(\cdot)$ and $\mathrm{c}(\cdot) = \cos(\cdot)$.

MPC Algorithm: Objective Function (cont)

The objective function is not convex.

MPC Algorithm: Objective Function (cont)

The objective function is not convex.

Hence the convex approximations

$$\cos(\hat{\theta}_{i,k}) \approx \cos(\theta_{i,k}^{disc}) - \sin(\theta_{i,k}^{disc})(\hat{\theta}_{i,k} - \theta_{i,k}^{disc})$$

and

$$\sin(\hat{\theta}_{i,k}) \approx \sin(\theta_{i,k}^{disc}) + \cos(\theta_{i,k}^{disc})(\hat{\theta}_{i,k} - \theta_{i,k}^{disc})$$

are necessary to make the optimization problem solvable.

Resulting Trajectory

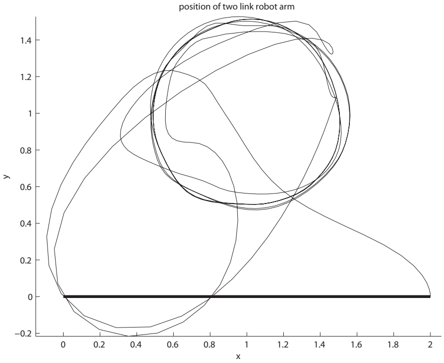

Figure: Trajectory I.

Resulting Trajectory

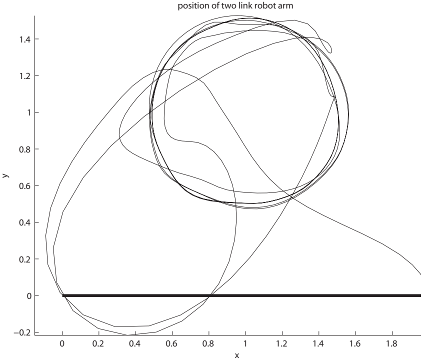

Figure: Trajectory I.

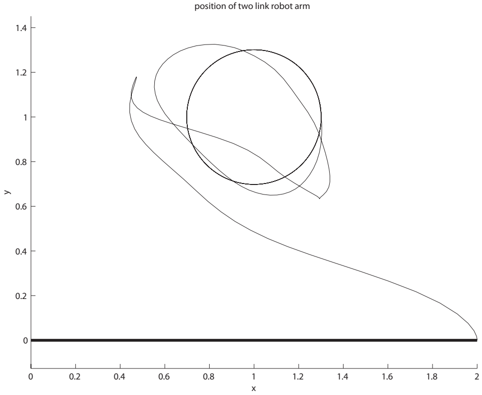

Figure: Trajectory II.

Comments

- The MPC was constructed from scratch.

Comments

- The MPC was constructed from scratch.
- It is in general difficult to formulate a nonlinear MPC problem as a convex problem.

Comments

- The MPC was constructed from scratch.
- It is in general difficult to formulate a nonlinear MPC problem as a convex problem.
- No guarantee of *stability*.

Comments

- The MPC was constructed from scratch.
- It is in general difficult to formulate a nonlinear MPC problem as a convex problem.
- No guarantee of *stability*.
- The MPC is *robust* for a small model mismatch.

Conclusion

- The method used works extremely well.

Conclusion

- The method used works extremely well.
- It was easy to implement.

Conclusion

- The method used works extremely well.
- It was easy to implement.
- It would be interesting to test the procedure on a real system.

Thank you