Deterministic Robot Motion Planning for Safety-critical Robot Applications

Torsten Kroeger
Introduction

- **Sensor integration** and safety on different motion control levels belong the keys for future advancements of **robotic systems**.
Introduction

- **Sensor integration** and safety on different motion control levels belong the keys for future advancements of **robotic systems**.

- In general, we distinguish between:
  1. Trajectory-following motions (without sensors)
  2. Sensor-guided motions
     - Force/torque control
     - Visual servo control
     - ...

- In practice, we need **both** (in certifiable implementations):

  ![Diagram](image.png)
Outline

Real-time robot motion planning

Deterministic responses to sensor events

Hybrid control

Deterministic switching

Stable and safe
Switching from sensor-guided to trajectory-following robot motion control
Input and Output Values

Target state of motion at instant $t_i$

Motion constraints at instant $t_i$

Motion state at instant $t_i$

Set-points for lower-level control

On-line trajectory generation algorithm

Type II
Input and Output Values

Target state of motion at instant $t_i$

Motion constraints at instant $t_i$

Motion state at instant $t_i$

Set-points for lower-level control

Motion state at instant $t_{i+1}$

Type V
Input and Output Values

Target state of motion at instant $t_i$

- $P_i^{trgt}$
- $V_i^{trgt}$
- $A_i^{trgt}$
- $J_i^{trgt}$
- $V_i^{max}$
- $A_i^{max}$
- $J_i^{max}$
- $D_i^{max}$
- $P_i$
- $V_i$
- $A_i$
- $J_i$

Motion constraints at instant $t_i$

Motion state at instant $t_i$

On-line trajectory generation algorithm

Set-points for lower-level control

Motion state at instant $t_{i+1}$

Type IX
The Basic Idea

There is a **finite set** of possible motion profiles, of which one transfers one single selected DOF from the initial state of motion $\vec{M}_i$ to its target state of motion $\vec{M}_i^{trgt}$ within the shortest possible time (time-optimally).

<table>
<thead>
<tr>
<th>$\forall i \in \mathbb{Z}$</th>
<th>$V_i^{trgt} = 0 \land A_i^{trgt} = 0 \land J_i^{trgt} = 0$</th>
<th>$V_i^{trgt} \in \mathbb{R} \land A_i^{trgt} = 0 \land J_i^{trgt} = 0$</th>
<th>$V_i^{trgt} \in \mathbb{R} \land A_i^{trgt} \in \mathbb{R} \land J_i^{trgt} = 0$</th>
<th>$V_i^{trgt} \in \mathbb{R} \land A_i^{trgt} \in \mathbb{R} \land J_i^{trgt} \in \mathbb{R}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_i^{max}$</td>
<td>$\in \mathbb{R}$</td>
<td>$\in \mathbb{R}$</td>
<td>$\in \mathbb{R}$</td>
<td>$\in \mathbb{R}$</td>
</tr>
<tr>
<td>$J_i^{max}$</td>
<td>$= \infty$</td>
<td>$= \infty$</td>
<td>$= \infty$</td>
<td>$= \infty$</td>
</tr>
<tr>
<td>$D_i^{max}$</td>
<td>$= \infty$</td>
<td>$= \infty$</td>
<td>$= \infty$</td>
<td>$= \infty$</td>
</tr>
</tbody>
</table>

> Velocity profiles

> Acceleration profiles

> Jerk profiles
Set of All Type IV Acceleration Profiles

- A system of nonlinear equations can be set up for every acceleration profile. The solution contains all parameters for the calculation of...
Set of All Type IV Acceleration Profiles

- Each system of equations is solvable for a proper subset of $\mathbb{R}^8$.

- The union of all these subsets constitutes the $\mathbb{R}^8$:

$$
\bigcup_{r=0}^{R} rD_{\text{Step1}} = \mathbb{R}^8.
$$
Decision Trees

**Input:**
- current state of motion
- target state of motion
- kinematic motion constraints

**Output:**
- acceleration profile

Decision Trees diagram with conditions for calculating the minimal execution time of a concrete acceleration profile.
Systems of Nonlinear Equations

- Example: \[ r \Psi_{Step1}^{Step1} = \text{PosTriNegTri} \]
\[ 2t_i - T_i = \frac{(a_{\text{peak}1} - A_i)}{J_{i}^{\text{max}}} \]  
\[ 3t_i - 2t_i = \frac{a_{\text{peak}1}}{J_{i}^{\text{max}}} \]  
\[ 4t_i - 3t_i = -\frac{a_{\text{peak}2}}{J_{i}^{\text{max}}} \]  
\[ t_{i}^{\text{min}} - 4t_i = -\frac{a_{\text{peak}2}}{J_{i}^{\text{max}}} \]  
\[ 2v_i - V_i = \frac{1}{2} (2t_i - T_i) (A_i + a_{\text{peak}1}) \]  
\[ 3v_i - 2v_i = \frac{1}{2} (3t_i - 2t_i) a_{\text{peak}1} \]  
\[ 4v_i - 3v_i = \frac{1}{2} (4t_i - 3t_i) a_{\text{peak}2} \]  
\[ V_{i}^{\text{trgt}} - 4v_i = \frac{1}{2} (t_{i}^{\text{min}} - 4t_i) a_{\text{peak}2} \]  
\[ 2p_i - P_i = V_i (2t_i - T_i) + \frac{1}{2} A_i (2t_i - T_i)^2 + \frac{1}{6} J_{i}^{\text{max}} (2t_i - T_i)^3 \]  
\[ 3p_i - 2p_i = 2v_i (3t_i - 2t_i) + \frac{1}{2} a_{\text{peak}1} (3t_i - 2t_i)^2 - \frac{1}{6} J_{i}^{\text{max}} (3t_i - 2t_i)^3 \]  
\[ 4p_i - 3p_i = 3v_i (4t_i - 3t_i) - \frac{1}{6} J_{i}^{\text{max}} (4t_i - 3t_i)^3 \]  
\[ P_{i}^{\text{trgt}} - 4p_i = 4v_i (t_{i}^{\text{min}} - 4t_i) + \frac{1}{2} a_{\text{peak}2} (t_{i}^{\text{min}} - 4t_i)^2 + \frac{1}{6} J_{i}^{\text{max}} (t_{i}^{\text{min}} - 4t_i)^3 \]
Sample Result for One DOF

$r\Psi_{\text{Step}} = \text{PosTriNegTri}$
The Reflexxes Motion Libraries

Target-position-based Library (Type IV)

Selection of degrees of freedom $\vec{S}_i$

Target position $\vec{P}_{i}^{\text{trgt}}$

Target velocity $\vec{V}_{i}^{\text{trgt}}$

Target state of motion

Maximum velocity $\vec{V}_{i}^{\max}$

Maximum acceleration $\vec{A}_{i}^{\max}$

Maximum jerk $\vec{J}_{i}^{\max}$

Kinematic motion constraints

Position $\vec{P}_{i}$

Velocity $\vec{V}_{i}$

Acceleration $\vec{A}_{i}$

Current state of motion

Control cycle $T_i$

Refluxxes Type IV Motion Library

Position $\vec{P}_{i+1}$

Velocity $\vec{V}_{i+1}$

Acceleration $\vec{A}_{i+1}$

New state of motion

Control cycle $T_{i+1}$
Reflexxes: Simple C/C++ API (2010+)

Constructor

```
ReflexxesAPI( const unsigned int &DegreesOfFreedom,
              const double &CycleTimeInSeconds,
              const unsigned int &NumberOfAdditionalThreads = 0,
              const double &MaxTimeForOverrideFilterInSeconds = 2.0);
```

Target position-based algorithm

```
int RMLPosition( const RMLPositionInputParameters &InputValues,
                 RMLPositionOutputParameters *OutputValues,
                 const RMLPositionFlags &Flags);
```

Target velocity-based algorithm

```
int RMLVelocity( const RMLVelocityInputParameters &InputValues,
                 RMLVelocityOutputParameters *OutputValues,
                 const RMLVelocityFlags &Flags);
```

Destructor

```
~ReflexxesAPI(void);
```
Reflexxes Motion Libraries

Open source software, tutorials, examples…

www.reflexxes.com
JediBot – A Student Project (2011)
JediBot – A Student Project (2011)
Telescope Control

Victor M. Blanco telescope, Santiago de Chile.

Collaboration with the Cerro Tololo Inter-American Observatory, Chile.
Telescope Control

Collaboration with the Cerro Tololo Inter-American Observatory, Chile.

Large Synoptic Survey Telescope, LSST (start of operation: 2020).
KUKA LBR iiwa – Online Trajectory Generation


www.kuka.com
Real-time robot motion planning

Deterministic responses to sensor events

Hybrid control

Deterministic switching

Stable and safe
Sensor integration and safety on different motion control levels belong the keys for future advancements of robotic systems.
Switched-system Control in Robotics

- Controllers cannot take over in any situation, e.g.,
  - no valid sensor signal,
  - force control without contact,
  - sudden changes of the environment,
  - sensor malfunctions, or
  - switching to a controller yields heavy jerk.

- Many compliant motion tasks require instantaneous switchings from one controller to another in order to achieve stability and guarantee safety.

- Requirement: stable and deterministic switchings!

- Motion specification by Primitives
The Adaptive Selection Matrix

Classical selection matrix

Adaptive selection matrix

An example:

‘Force in z-Direction’

<table>
<thead>
<tr>
<th>Level 0</th>
<th>Level 1</th>
<th>Level 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force</td>
<td>Vision</td>
<td>Velocity</td>
</tr>
</tbody>
</table>
Task Space Control Scheme

- Adaptive selection matrix
- Target-position-based OTG
- Target-velocity-based OTG
- Force controller
- Visual servo control module
- Distance controller
- Set points and parameters for position-controlled DOFs
- Set points and parameters for speed-controlled DOFs
- Calculation of cart. pose set point
- Transf. to joint space
- Joint controller
- Robot and environment
- User exit condition
- Jacobian
- Forward kinematics
- Distance sensor
- Camera system
- Force/torque sensor
- Distance set point
- Vision set points
- Force/torque set points
- Manipulation primitive parameters
Control Scheme

- Hybrid switched-control (simplified for one DOF)
Industrial Safety Standards

- NUM EtherCAT Terminals
- NUM Safe PLC & Safe Terminals
- NCK Network Multi-NCK architecture
- Axes and Spindle Motors

FSoE (Fail Safe over EtherCAT)

Operating Panel PC

- EtherCAT
- CANopen

HBA-X

CANopen Axes

nPad Portable Handheld Panel

Machine Panel

RTEthernet

NUM Servobus

NCK 1

NCK 2

NCK n

Safe Axes

Interpolated Axes
Up to 32 axes per NCK and more than 200 per CNC System

Host

Ethernet TCP/IP
Industrial Safety Standards

NUM EtherCAT Terminals
NUM Safe PLC & Safe Terminals

EtherCAT

EtherCAT

CANopen

CANopen Axes

HBA-X

nPad
Portable Handheld Panel

Operating Panel PC

RTEthernet

NUM Servobus

NCK Network
Multi-NCK architecture

Axe and Spindle Motors

NCK 1

NCK 2

NCK n

Host

Interpolated Axes
Up to 32 axes per NCK and more than 200 per CNC System
Industrial Safety Standards

Master B
NUM EtherCAT Terminals
NUM Safe PLC & Safe Terminals
 EtherCAT
HBA-X
 CANopen
nPad
Portable Handheld Panel

Master A
EtherCAT
 CANopen
 CANopen Axes
 Operating Panel PC
 Machine Panel
 nPad
Portable Handheld Panel
 Interpolated Axes
Up to 32 axes per NCK and more than 200 per CNC System

NCK Network
Multi-NCK architecture
 Axes and Spindle Motors
 NUM Servobus
 NCK 1
 NCK 2
 NCK n
 Ethernet
TCP/IP
 Host

Copyright 2012 Wordsun.com..
Consistency During Switching

$t = T_i = 0$

\[ \vec{M}_i \]

\[ \vec{M}_i^{tgt} \]
Consistency During Switching

\[ t = T_i = 0 \]
Consistency During Switching

\[ t = T_{i+1} \]

\[ \vec{M}_i^{trgt} \]
Consistency During Switching

\[ t = T_{i+2} \]

\[ \overrightarrow{M}^{tgt}_i \]

Master B
Consistency During Switching

\[ t = T_{i+3} \]

\[ \overrightarrow{M}_{i}^{trgt} \]

Master B
Sensor Failures
Hybrid Switched-System Control
• Stabilizing hybrid switched-systems (velocity target)

Velocity-acceleration plane in state space:

\[ k_v(t) \forall k \in \{x, y, z\} \text{ in } \circ/s \]
Hybrid Switched-System Control

Set-point signals

- Force torque controller
- Visual servo controller
- Distance controller

Open-loop position controller (OTG)
Open-loop velocity controller (OTG)

Switching signal \( \sigma \)

Supervisory decision maker

External data/signals/parameters

Actuator controller

Actuator, sensors, and environment

\( \vec{u}^{(1)} \rightarrow \vec{u}^{(2)} \rightarrow \vec{u}^{(3)} \rightarrow \ldots \rightarrow \vec{u} \)

\( \vec{y} \)
• Stabilizing hybrid switched-systems

Velocity-acceleration plane in state space:

\[ k_v(t) \forall k \in \{x, y, z\} \text{ in } \frac{\text{in/s}}{\text{s}} \]

\[ k_a(t) \forall k \in \{x, y, z\} \text{ in } \frac{\text{mm/s}^2}{\text{s}} \]

Instability or potential instability detected at \( t = 663 \text{ ms} \)
Conclusion

- The proposed framework of online trajectory generation allows **instantaneous switching** in case of controller and sensor failures.

- This includes **sudden changes of controller gains** and taking over from other controllers in **stable** and **safe** manner.

- The framework work also allows implementing **redundant system**, between which can be switched in case of failures.
Thank you!

torsten@kit.edu