Basic tutorial 01.01.002 - Working with the class iU_gearPairDesignTarget

created by: Florian Eigner | imk Health Intelligence GmbH | Amselgrund 30 | 09128 Chemnitz

contact: florian.eigner@imk-ic.com

last modified: 01.04.2022

Time to read: approx. 15 min **Time to practice:** 0 min

Table of Contents

1 Intention of this tutorial

This tutorial introduces the basics of the class **iU_gearPairDesignTarget.** You will learn how to set the parameters for different types of gearing, wheras the point of action (contact point for initial kinematic calculation) is always located on a straight line between the footpoints of both axis.

Note: This tutorial contains no exercices. For understanding the influence on the createGearPair process you need more information to new classes than this tutorial provides. To practice the behavior for different **iU_gearPairDesignTarget** settings go to tutorial_01_01_003_createGearPair.

2 New classes in this tutorial

2.1 class iU_gearPairDesignTarget

2.1.1 Parameters

There are two mandatory parameters for proper use:

- ***name*** → the name of the feature, which controlles the appearance of an gearPair. This can be one of the following options: *'radiusRatio', 'pitch', 'betaSource' or 'betaTarget'.*

- ***value*** → the value defines the value of the propertie ***name.*** Within the **createGearPair()** process a point of action is calculated to reach this key-value-pair.

Note: It is possible to create an instance of **iU_gearPairDesignTarget** without any ***name/value*** pair. In this case within the **createGearPair()** a ***num_radiusRatio***. is calculated depending on the transmittion ratio of the new created **clc_gearPair.**

To control the basic **createGearPair()** process when a certain pitch should be created one argument is optional:

- ***secondaryArgument*** → chooses one of two solutions, if ***name*** = *'pitch'* .

2.1.2 Evaluation

2.1.3 Influence on the calculation process

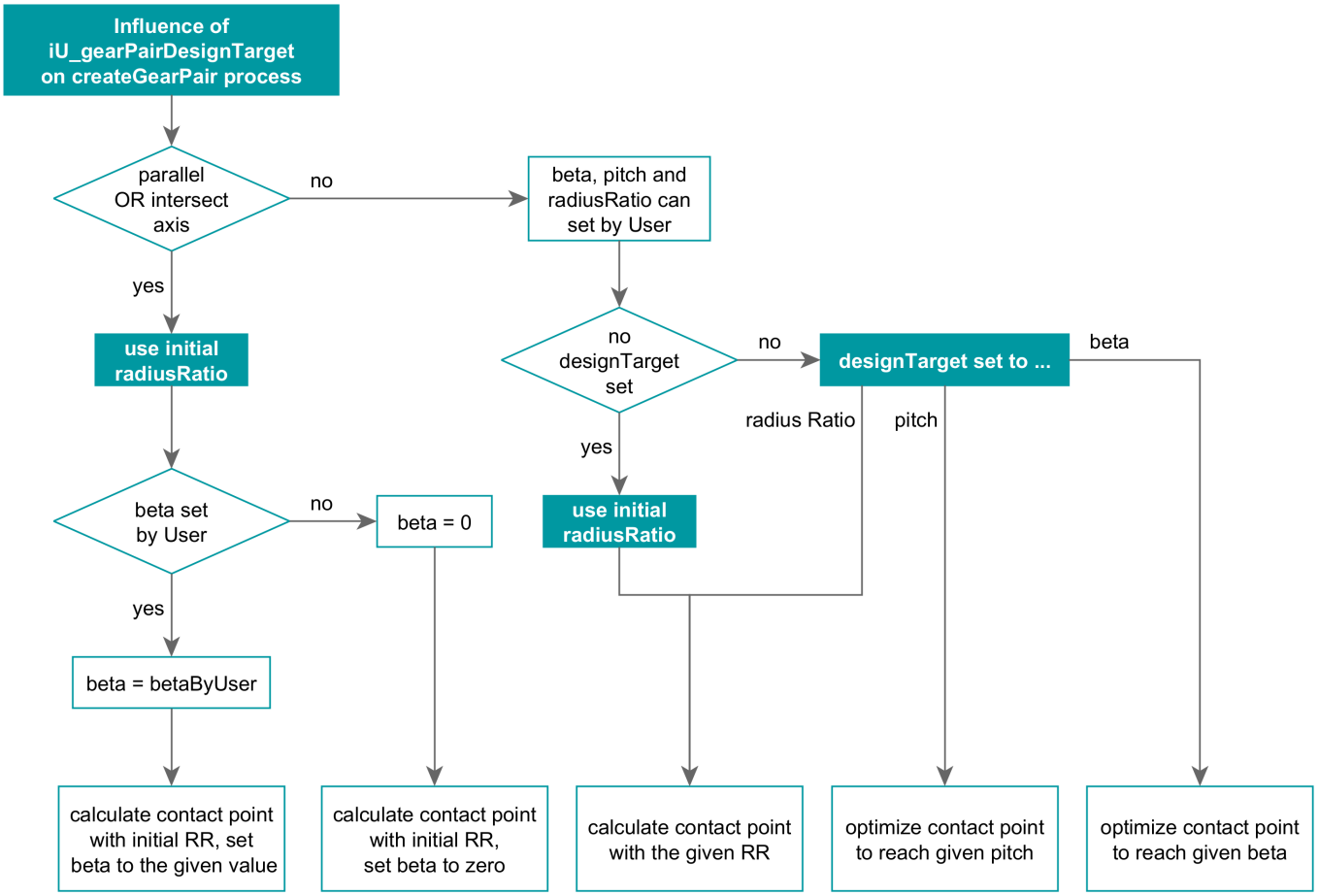
This overview shows, which settings in the class **iU_gearPairDesignTarget** lead to which creation processes in the ***createGearPair*** method. Basic tutorial tutorial_01_01_003_createGearPair give more information and excercises about the creation process. The initial radiusRatio is defined by

$$radiusRatio = -1\frac{cos(\delta_{axis}) - i_{21}}{cos(\delta_{axis}) - \frac{1}{i_{21}}}$$ with

- δ_{axis} ... crossing angle of the axis ... property of class **iU_gearPairStatics** (***num_crossingAngle***)
- i_{21}... transmission ratio of the gear pair ... property of class **iU_gearPairStatics** (***num_i21***)

More information: tutorial_01_01_003_createGearPair

This formula can be derived from Klingelnberg, page 41, formula 2.6 ... exact source to be added.



3 Example

3.1 Initialize with no additional arguments

With this option, initialize with no additional arguments, you create a iU_gearPairDesignTarget, that leads in the createGearPair process to a gearPair with the initial radiusRatio.

```
thisDesignTarget = iU_gearPairDesignTarget()
```

```
thisDesignTarget =
  iU_gearPairDesignTarget with properties:

                name: [1×0 char]
               value: 0
    secondaryArgument: 1
         axialRefGear: 'source'
        axialRefPoint: 'footPoint'
     axialUseMidPoint: 0
          axialOffset: 0
               shape: 'hyperbolic'
```

3.2 initialize with radiusRatio or any beta angle

```
thisDesignTarget = iU_gearPairDesignTarget('radiusRatio', -0.78)
```

```
thisDesignTarget =
  iU_gearPairDesignTarget with properties:

                name: 'radiusRatio'
               value: -0.7800
    secondaryArgument: 1
```

```
          axialRefGear: 'source'
         axialRefPoint: 'footPoint'
      axialUseMidPoint: 0
           axialOffset: 0
                 shape: 'hyperbolic'
```

Its also possible to initialize without any arguments and set those arguments later

```
thisDesignTarget = iU_gearPairDesignTarget();
thisDesignTarget.name = 'betaSource';
thisDesignTarget.value = 20 / 180*pi; % / 180 * pi to get the radiant value for
20 degree
thisDesignTarget
```

```
thisDesignTarget =
  iU_gearPairDesignTarget with properties:

                 name: 'betaSource'
                value: 0.3491
    secondaryArgument: 1
         axialRefGear: 'source'
        axialRefPoint: 'footPoint'
     axialUseMidPoint: 0
          axialOffset: 0
                shape: 'hyperbolic'
```

3.3 Initialize with pitch

The calculation process with a given pitch can lead to no or two solutions. In the case of no solution the requested pitch is greater than the maximal pitch. For this maximum pitch there is a specific ***radiusRatio*** called $\mathrm{rR\_forMaxPitch}$. In the case, the ***secondaryArgument*** decides on the choice of one solution by the following search intervall ...

```
% secondaryArgument = \left\{
% \begin{array}{l}
% 1 & \rightarrow radiusRatio \in [rR_{maxPitch} , 100 \cdot rR_{maxPitch}]\\
% -1 & \rightarrow radiusRatio \in [rR_{maxPitch} , 0 \cdot rR_{maxPitch}]\\
% \end{array}
```

$$radiusRatio = \begin{cases} \in [rR_{maxPitch}, 100 \cdot rR_{maxPitch}] & if \quad secondaryArgument = 1 \\ \in [rR_{maxPitch}, 0 \cdot rR_{maxPitch}] & if \quad secondaryArgument = -1 \end{cases}$$

... to minimize an error function. The initial value for ***secondaryArgument*** is ***1.***

Note: If the given pitch is greater than the maximum pitch, there is no solution. You must reduce the pitch.

Initilize the **iU_gearPairDesignTarget** with a modulus (i.e.) of $m_n = 2$. Using $\mathrm{pitch} = m_n \cdot \pi$ leads to the following constructor:

```
thisDesignTarget = iU_gearPairDesignTarget('pitch', 2 * pi);
```

To choose one of the two possible solutions an ***secondaryArgument*** is needed.

```
thisDesignTarget.secondaryArgument = -1;
thisDesignTarget
```

```
thisDesignTarget =
```

```
  iU_gearPairDesignTarget with properties:

                  name: 'pitch'
                 value: 6.2832
    secondaryArgument: -1
         axialRefGear: 'source'
        axialRefPoint: 'footPoint'
     axialUseMidPoint: 0
          axialOffset: 0
                 shape: 'hyperbolic'
```

```
disp('END OF TUTORIAL')
```

```
END OF TUTORIAL
```