

Makroverarbeitung im XCTL-System

David Damm

28th January 2004

Aufbau der Makrodateien

Makrodatei = { *Makro* '\n' }+

Makro = *Commonblock* '\n' *Befehlsblock*

Commonblock = '[Common]' '\n' *Namensdefinition*
 '\n' *Längendefinition*

Namensdefinition = 'Name' *Trennzeichen* *Makroname*

Längendefinition = 'Length' *Trennzeichen* *Zahl*

Makroname = 'AreaScanJob' | 'AzimutalJustify' | 'InquireHwb' |
 'ScanJob' | 'SearchReflection' | 'SetupTopography' |
 'Test'

Zahl = '1' | '2' | '3' | ...

Befehlsblock = '[Commands]' '\n' *Befehlsfolge* '[End]'

Befehlsfolge = { *Befehl* {*Trennzeichen* *Parameter*}* { *Wert* }*'\n' }+

Befehl = 'AreaScan' | 'Calculate' | 'ChooseAxis' |
 'ChooseDetector' | 'ControlFlank' |
 'GotoIntensity' | 'GotoPeak' | 'LoadPoint' |
 'MoveToPoint' | 'Stop' |
 'SaveData' | 'Scan' | 'SetFileName' |
 'SetupScan' | 'SetWidth' | 'ShowValue'

Parameter = 'LargeSide' | 'SmallSide' | 'ToSmallerAngle' |
 'Interpolation' | 'LastGoal' | 'Standard' |
 'Result' | 'Argument' | 'Hwb' | 'Difference' |
 'Opposite' | 'ToLargerAngle' | 'BackMove' |
 'Middle' | 'ForScan' | 'ForAreaScan'

Befehle

Alle Befehle haben ein festgelegtes Format, das sich im allgemeinen wie folgt darstellen läßt:

```
Befehl Parameter1 Parameter2 Parameter3
```

AreaScan

AreaScan { ForScan | ForAreaScan } [StandardScan | Omega2ThetaScan] .

Calculate

Calculate { Difference | Opposite | Hwb | Middle } .

1. Difference

Hier wird die Differenz zwischen zwei Werten berechnet, die mit `LoadPoint` zwischengespeichert wurden. Das Ergebnis wird dann nach der Formel

$$\text{Result} = \text{Argument2} - \text{Argument1}$$

berechnet.

2. Opposite

Bei Angabe dieses Parameters, wird der Peak hergenommen und das erste Argument davon abgezogen.

$$\text{Result} = 2 * \text{PeakPoint} - \text{Argument1}$$

3. Hwb

Es wird die Halbwertsbreite ermittelt. Die Halbwertsbreite wird zunächst wie bei Difference berechnet.

$$\text{Hwb} = \text{Argument2} - \text{Argument1}$$

Ist die geforderte Einheit in Grad oder Minuten angegeben, so erfolgt noch eine zusätzliche Umrechnung der Halbwertsbreite.

$$\text{Hwb} = 60 * \text{Hwb}$$

4. Middle

$$\text{Result} = (\text{Argument1} + \text{Argument2}) / 2$$

ChooseAxis

ChooseAxis { DC | DF | CC | AR | TL | Omega | Phi |
Psi | Theta | X | Y | Z | Absorber } .

ChooseDetector

ChooseDetector { Counter | PSD } [Time Counts Fl] .

ControlFlank

```
ControlFlank { SmallSide | LargeSide } ControlRange .
```

GotoIntensity

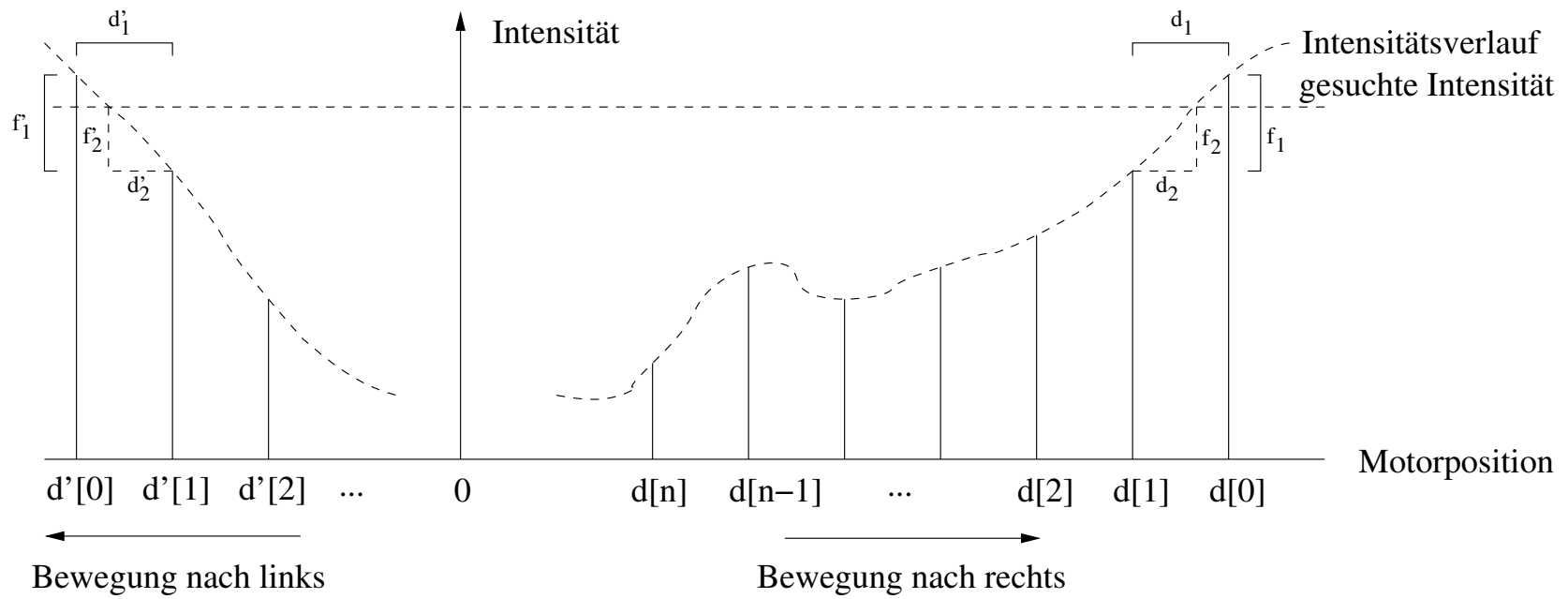
Mit dem Befehl `GotoIntensity` kann man eine bestimmte Zielintensität relativ zur Peakintensität anfahren:

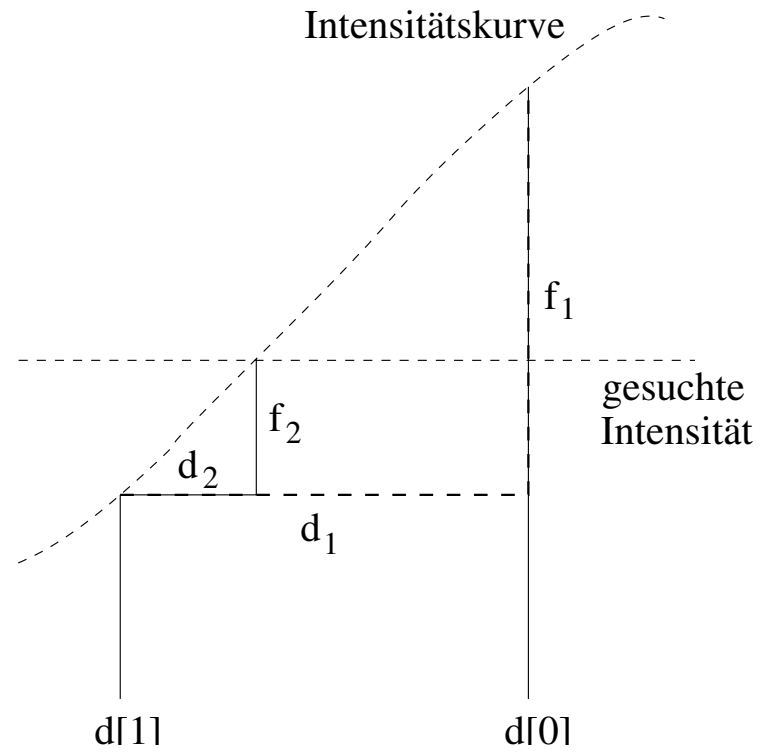
```
GotoIntensity { SmallSide | LargeSide }  
              { Interpolation | BackMove } Value .
```

Der erste Programmparameter gibt die Bewegungsrichtung an, entweder nach links (`SmallSide`) oder nach rechts (`LargeSide`).

Bei `BackMove` wird der Motor um einen Schritt zurückgefahren. Bei der `Interpolation` wird die anzufahrende Position aus den letzten beiden Positionen ermittelt.

Im dritten Parameter `Value` wird ein Wert angegeben: $\text{Zielintensität} = \text{Value} * \text{Peakintensität}$.





Nach dem Strahlensatz gilt dann

$$\frac{d_1}{f_1} = \frac{d_2}{f_2}$$

$$\begin{aligned} \Rightarrow d_2 &= \frac{d_1 \cdot f_2}{f_1} \\ &= \frac{(d[0] - d[1]) \cdot (fSearchIntensity - f[1])}{f[0] - f[1]}. \end{aligned}$$

Beispiel: sei $d[1] = 10$, $d[0] = 11$, $f[1] = 900$, $f[0] = 1200$, $fIntensitySearch = 1000$.

$$d_2 = \frac{(11 - 10) \cdot (1000 - 900)}{1200 - 900} = \frac{1}{3}.$$

Die neue Position ergibt sich dann als

$$pos = d[1] + d_2 = 10 + \frac{1}{3}.$$

GotoPeak

```
GotoPeak { ToSmallerAngle | ToLargerAngle }  
         { Interpolation | BackMove} .
```

LoadPoint

```
LoadPoint { Start | Peak } .  
LoadPoint Argument Value .
```


MoveToPoint

MoveToPoint { Result | Peak | Start | LastGoal } .
MoveToPoint Relative Value .
MoveToPoint Value .

SaveData

SaveData { ForScan | ForAreaScan } .

Scan

Scan { ForScan | ForAreaScan } .

SetFileName

SetFileName Name .

SetupScan

```
SetupScan { StandardScan | Omega2ThetaScan }  
ScanMin ScanWidth ScanMax .
```

SetWidth

SetWidth Value .

ShowValue

ShowValue { Hwb | Start | Peak } .