

## **Motion Control Praxis**

<b>PID-Regelung optimieren</b>	<b>2</b>
So funktioniert der Regelprozess	2
Bedeutung und Einfluss der Reglerparameter	2
PID Faktoren	4
Schritt für Schritt die Reglerparameter optimieren	6
<b>Kurvenscheiben-Steuerung (CAM)</b>	<b>9</b>
Schnellkurs für Ungeduldige	10
Beispiel: Kartons mit Haltbarkeitsdatum stempeln	10
Beispiel: Kartons bedrucken mit Markerkorrektur	13
Beispiel: Slave-Synchronisation mit Marker	17
<b>Nockenschaltwerk</b>	<b>20</b>
Beispiel eines Nockenschaltwerks	20
<b>Ruckbegrenzung</b>	<b>21</b>
Beispiele für ruckbegrenzte Bewegungen	23

### **PID-Regelung optimieren**

Das **Tune Oszilloskop** können Sie benutzen, um die Steuerungsparameter zu optimieren und damit eine bestmögliche Leistung des Systems zu erreichen. Dazu müssen Sie nur einige Dinge über den Regelprozess von APOSS wissen. Außerdem muss dazu der Motor bzw. die Anwendung fahren können.

### **So funktioniert der Regelprozess**

Die in der Positioniersteuerung integrierte Lagereglereinheit besteht aus zwei Teilen:

1. Der Sollwert-Generator wertet die verschiedenen Positionierbefehle aus und erzeugt eine Reihe von Sollpositionen, die schon in der gewünschten Position enden könnten.  
Alle Positioniervorgänge weisen normalerweise einen trapezförmigen Geschwindigkeitsverlauf auf. Das bedeutet, dass nach einer Phase mit konstanter Beschleunigung eine Phase mit konstanter Geschwindigkeit und zuletzt wiederum eine Phase mit konstanter negativer Beschleunigung folgt, die in der gewünschten Zielposition endet.
2. Die PID-Regelung erhält die Sollpositionen vom Sollwert-Generator und berechnet in Zyklen den Drehzahlsollwert, der für den Motor benötigt wird, damit er die aktuelle Sollposition erreicht.  
Mit den Reglerparametern können Sie direkt beeinflussen, wie stark und wie schnell einer Abweichung von dem theoretischen Sollverlauf entgegengewirkt werden soll.

Folgende Anzeichen deuten darauf hin, dass die Reglerparameter nicht optimal eingestellt sind:

- Antrieb schwingt
- Antrieb ist sehr laut
- häufiges Auftreten von Schleppfehlern
- schlechte Regelgenauigkeit

**!!!** Die Reglerparameter sind lastabhängig. Daher sollte der Antrieb unter den tatsächlichen Einsatzbedingungen optimiert werden.

In Ausnahmefällen kann es bei stark schwankenden Lastverhältnissen notwendig sein, dass man verschiedene Sätze von Reglerparametern ermittelt und dann im späteren Anwendungsprogramm in Abhängigkeit vom Fahrvorgang entsprechend umprogrammiert.

### **Bedeutung und Einfluss der Reglerparameter**

Die Lagereglereinheit der Positioniersteuerung überträgt den aktuell benötigten Motorstrom über einen analogen Sollwert an den Verstärker. Entsprechend dieser Sollwertvorgabe regelt der Verstärker den Motorstrom und somit das Motordrehmoment.

Dieser analoge Sollwert wird periodisch jede Millisekunde neu berechnet (das Intervall kann mit dem Parameter TIMER programmiert werden).

APOSS ist mit – für die meisten Anwendungen – passenden Reglerparametern vorbereitet.

Die PID-Regelung arbeitet nach folgender Formel:

$$1 = (\text{FFVEL normiert}) * (\text{Sollgeschwindigkeit CV})$$

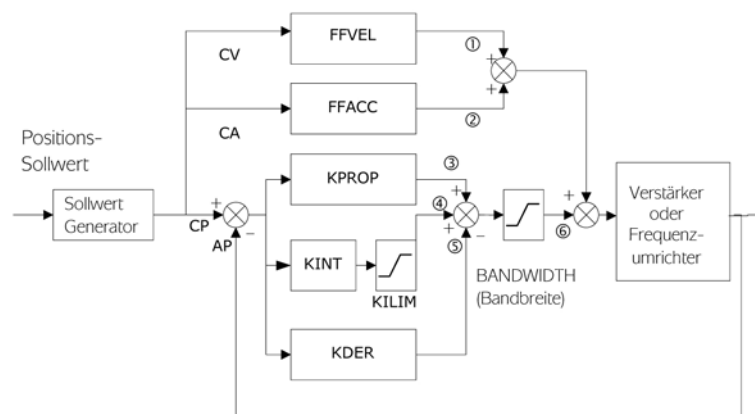
$$2 = (\text{FFACC normiert}) * (\text{Sollbeschleunigung CA})$$

$$3 = \text{KPROP} * (\text{Positionsabweichung CP-AP})$$

$$4 = \text{KINT} * (\text{Summe aller vorhergehenden Positionsabweichungen}) \text{ (begrenzt durch KILIM) normiert}$$

$$5 = \text{KDER} * (\text{Differenz der Positionsabweichung})$$

$$6 = 3 + 4 + 5 \text{ (begrenzt durch BANDWIDTH)}$$



AV ist die aktuelle Geschwindigkeit; sie wird ermittelt als Differenz der Ist-Geschwindigkeit minus der letzten Ist-Geschwindigkeit;

AP ist die aktuelle Position (berechnet durch den Drehgeber-Istwert) in qc;

CP ist die aktuelle Sollposition in qc;

CV ist die Sollgeschwindigkeit in qc/ms; sie wird ermittelt als Differenz der aktuellen Sollposition minus der letzten Sollposition;

CA ist die Sollbeschleunigung.

Positionsabweichung = wird durch  $CP - AP$  berechnet

Differenz der Positionsabweichung = Positionsabweichung – alte Positionsabweichung

KILIM normiert = bei einem KILIM Wert von 1000 wird ein Sollwert von 100 % erreicht.

FFVEL normiert = bei einem FFVEL Wert von 1000 und einer Sollgeschwindigkeit == VELMAX wird ein Sollwert von 100 % erreicht

FFACC normiert = bei einem FFACC Wert von 1000 und einer Sollbeschleunigung == VELMAX/ms entsteht ein Sollwert von 100 %.

!!! Im Modus SYNCV arbeitet die PID-Regelung mit einer Drehzahl- statt einer Positions-änderung. Die Drehzahländerung wird durch  $CV - AV$  berechnet.

Die Regelung nutzt zwei Strategien gleichzeitig:

1. Eine Feedforward-Regelung mit offenem Regelkreis.  
Da ein asynchroner Motor schon an sich eine gute Leistung im offenen Regelkreis hat, ist für die meisten Anwendungen die Feedforward-Regelung ein wichtiger Teil der Steuerung. Die Benutzung der Feedforward-Regelung fördert eine sehr schnelle und exakte Reaktion auf die Änderungen der Sollposition.
2. Eine PID-Regelung mit geschlossenem Regelkreis.  
Die PID-Regelung überwacht die Differenz zwischen der aktuellen Position und der Sollposition. Basierend auf diesen Informationen berechnet die PID-Regelung einen Sollwert um die Positionsabweichung zu minimieren. So können Änderungen der Last oder Reibung ausgeglichen werden.  
Die PID-Regelung ist auch notwendig, um mögliche Positionsabweichungen auszugleichen, die durch eine ungenaue Einstellung der Feedforward-Regelung mit offenem Regelkreis verursacht werden.

Die Feedforward-Regelung wird eingesetzt, um Änderungen der Sollpositionen (besonders wichtig für Synchronisations-Anwendungen) zu handhaben, während die PID-Regelung benutzt wird, um Änderungen der Lastbedingungen oder Ungenauigkeiten der Feedforward-Regelung auszugleichen.

### PID Faktoren

Proportionalfaktor KPROP (11) Der **Proportionalfaktor** KPROP wird mit der Positionsabweichung multipliziert und das Ergebnis zum Sollwert (dem internen Sollwert für den Verstärker bzw. Motor) addiert. Da der berechnete Sollwert proportional zur Positionsabweichung (oder zum Positionsfehler) ist, wird diese Art der Regelung proportionale Regelung genannt.

Das Verhalten einer solchen proportionalen Regelung ähnelt dem einer Feder, die je stärker sie gedehnt wird, eine umso stärkere Gegenkraft erzeugt.

KPROP zu klein große Positionsabweichung durch nicht kompensierbare Last- und Reibungsmomente;

KPROP größer schnellere Reaktion, kleinere statische Positionsabweichung, stärkeres Überschwingen, geringere Dämpfung;

KPROP zu groß starke Schwingungen, Instabilität;

Differentialfaktor KDER (12) Der **Differentialfaktor** wird mit dem Differential der Positionsabweichung (der „Geschwindigkeit“ der Positionsabweichung) multipliziert und das Ergebnis zum Sollwert addiert. Das bedeutet umso stärker sich der Positionsfehler von Abtastzeitpunkt zu Abtastzeitpunkt verändert, umso größer ist der Differentialanteil.

Das Verhalten einer solchen Regelung ähnelt dem eines Dämpfungselements, das je schneller es verstärkt wird, eine umso stärkere Gegenkraft erzeugt. Daher erhöht der **Differentialfaktor** die Dämpfung im System.

KDER klein keine Wirkung;

KDER größer bessere Dämpfung, geringeres Überschwingen; falls gleichzeitig KPROP erhöht wird: schnellere Reaktion auf Regelabweichung bei gleich starken Schwingungen;

KDER zu groß starke Schwingungen, Instabilität;

Wenn man mit einem normalen PID-Algorithmus arbeitet, muss FFVEL (36) immer genau so groß wie der KDER-Anteil sein, um diese typische D-Dämpfung zu erreichen.

Wenn man jedoch träge große Motoren regelt, die schwingen könnten, wird man die Regelung in der Bandbreite begrenzen. Dazu setzt man BANDWIDTH (35) sehr klein, z.B. 10 % und arbeitet mit FFVEL (36) und FFACC (37), die beide relativ groß eingestellt werden.

Integalfaktor KINT (13) Die Summe aller Fehler wird jedes Mal berechnet, wenn der Sollwert erneuert wird. Der **Integalfaktor** wird dann mit der Summe aller Positionsfehler multipliziert und zum gesamten Sollwert addiert. Das bedeutet, dass der Integralanteil umso größer wird, je länger Positionsfehler mit gleichem Vorzeichen auftreten. Durch dieses Verhalten wird ein statischer Positionsfehler, der auf Grund einer konstanten Motorlast auftreten kann, zu Null ausgeregelt. Stellen Sie daher sicher, dass Sie den **Integalfaktor** anwenden, wenn in Ihrer Anwendung statische Positionsabweichungen auftreten.

Durch das **Integrationslimit** kann der maximale aufaddierte Regelfehler begrenzt werden (anti-wind-up).

KINT sehr klein statische Positionsabweichung wird sehr langsam zu Null ausgeregelt;

KINT größer schnellere Regelung der statischen Positionsabweichung zu Null, stärkeres Überschwingen;

KINT zu groß starkes Schwingen, Instabilität.

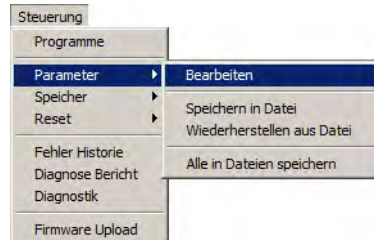
- Integrationslimit KILIM (21) Das **Integrationslimit** begrenzt den Sollwert, der durch den Integralfaktor erzeugt wird. Dadurch lässt sich das so genannte „wind-up“-Problem verhindern, das typischerweise bei Anwendungen auftritt, bei denen der gesamte Sollwert (der interne Drehzahlsollwert) so groß wird, dass es lange Zeit dauert, bis er wieder heruntergeregelt werden kann. So kann man verhindern, dass ein großer aufaddierter Positionsfehler zu einer zu starken Gegenreaktion führt, die das Antriebssystem zum Schwingen bringen könnte.
- Geschwindigkeits-Feedforward FFVEL (36) Der Parameter **Geschwindigkeits-Feedforward** ist ein Skalierungsfaktor, der mit dem Differential der Sollposition (der Geschwindigkeit der Sollposition) multipliziert wird. Das Ergebnis wird zum gesamten Sollwert addiert.
- Dieser Faktor eignet sich besonders bei solchen Anwendungen, bei denen es eine gute Wechselbeziehung zwischen dem Sollwert und der Drehzahl des Motors gibt. Und das ist in den meisten Anwendungen der Fall.
- Mit einem Wert von 1000 (empfohlen für die meisten Anwendungen) wird ein Sollwertanteil von 100 % erzeugt, wenn die Geschwindigkeit der Sollposition genauso groß ist wie das in den Parametern VELMAX (1) und ENCODER (2) gesetzte Maximum.
- !!! Wenn eine Regelung in der Bandbreite begrenzt ist, muss eine Grundgeschwindigkeit vorgegeben werden, damit ausgeschlossen wird, dass die Regelung durch die eingestellte Begrenzung das Fahren des Antriebs gänzlich verhindert.
- Beim Arbeiten mit einer normalen PID-Regelung muss FFVEL immer dieselbe Größe wie der KDER-Anteil haben, um eine typische KDER-Dämpfung zu erreichen.
- Beschleunigungs-Feedforward FFACC (37) Der Faktor **Beschleunigungs-Feedforward** wird mit dem zweiten Differential der Sollposition (die Beschleunigung der Sollposition) multipliziert und das Ergebnis zum Sollwert addiert. Dieser Faktor sollte benutzt werden, um den Drehmoment auszugleichen, der zum Beschleunigen und Bremsen des Trägheitsmoments benutzt wird.
- !!! Die Skalierung des FFACC Faktors hängt von der eingestellten **Kürzesten Rampe** RAMPMIN (31) ab. Sie sollten deshalb den FFACC Faktor entsprechend vergrößern, wenn Sie den Parameter **Kürzeste Rampe** verringern und umgekehrt.
- Geben Sie eine Grundbeschleunigung vor, wenn Sie die Regelung in der Bandbreite begrenzt haben. Damit verhindern Sie, dass die Regelung durch die eingestellte Begrenzung überhaupt nicht beschleunigt. Bei einer normalen PID-Regelung beträgt dieser Wert 0.
- Abtastzeit für PID-Regelung TIMER (14) Für besonders träge Systeme können Sie das gesamte Regelsystem verlangsamen, indem Sie für die Abtastzeit Vielfache von 1 ms eingeben. Allerdings gilt zu beachten, dass eine solche Änderung Einfluss auf sämtliche Reglerparameter hat! Daher sollte normalerweise in Parameter TIMER (14) nicht von dem Wert 1 ms abgewichen werden.
- BANDWIDTH (35) Eine **Bandbreite** von 1000 bedeutet, dass die Sollwerte zu 100 % ausgeführt werden, also **Differential-, Proportional-** und **Integralfaktor** wie definiert wirken.
- Wenn Sie aber ein schwingungsgefährdetes System betreiben, zum Beispiel einen Kran mit schweren Lasten, können Sie die Bandbreite in welcher die PID-Regelung wirken soll begrenzen. Ein BANDWIDTH von 300 bewirkt zum Beispiel eine Begrenzung auf 30 %: Das Aufschaukeln einer Schwingung wird dadurch verhindert, dass die Regelung nur mit 30 % des berechneten Sollwertes ausgeführt wird. Allerdings müssen Sie dann die Feedforward-Anteile benutzen, um eine entsprechende Regelung zu erreichen. Ein so eingestelltes System ist zwar nicht mehr so dynamisch, dafür aber wesentlich stabiler und es neigt weniger zu unkontrollierten Schwingungen.

### Schritt für Schritt die Reglerparameter optimieren

Am besten nutzen Sie dazu auch die Funktionen des **Tune Oszilloskops** es hilft durch die grafische Darstellung der Soll- und Istkurven die PID-Regelung zu beurteilen und zu optimieren.

Sie sollten jedoch immer nur einen Wert ändern und die Verbesserung mit Test im **Tune Oszilloskops** ermitteln.

Klicken Sie auf **Steuerung → Parameter → Achsen** und wählen Sie die Steuerung aus, den Sie gerade einstellen.



### Regelverhalten festlegen

Bevor Sie die Reglerparameter anpassen, legen Sie fest, welches Regelverhalten erzielt werden soll.

!!! Die Antriebselemente dürfen auf keinen Fall außerhalb der technischen Spezifikation betrieben werden; daher wird die Maximalbeschleunigung vom „schwächsten“ Antriebselement bestimmt.

- „Steife“ Achse: Eine möglichst schnelle Reaktion wird hauptsächlich durch den **Proportionalfaktor** beeinflusst. Das Ergebnis beurteilen Sie anhand der Geschwindigkeitsgrafik.
- Die Dämpfung der Schwingungen wird hauptsächlich durch den **Differentialfaktor** beeinflusst. Das Ergebnis ist am besten in der Geschwindigkeitsgrafik zu beurteilen.
- Eine kurzzeitige (statische) Positionsabweichung wird hauptsächlich durch den **Integralfaktor** reduziert und am besten anhand der Positionsgrafik beurteilt.

### Der schnellste Weg zur optimalen Regelung

Folgende vereinfachte Vorgehensweise führt in den meisten Fällen zu einem akzeptablen Ergebnis:

1. Setzen Sie den **Integrationsfaktor** KINT (13) auf null.
2. Erhöhen Sie den **Proportionalfaktor** KPROP (11) in Schritten, bis ein leichtes Überschwingen in der Geschwindigkeitsgrafik sichtbar wird.
3. Erhöhen Sie jetzt den **Differentialfaktor** KDER,(12), dadurch wird das Überschwingen wieder verringert.
4. Wiederholen Sie die Schritte 2 und 3 solange, bis keine Schwingungsneigung mehr auftritt und der Geschwindigkeitsverlauf dem gewünschten Regelverhalten angepasst ist.

### Zehn Schritte zur optimalen Regelung ... für M1 und neuere Steuerungen

Folgende Vorgehensweise optimiert die Steuerung für die meisten Anwendungen mit einer Motorsteuerung **M1** oder neueren Steuerungen:

1. Stellen Sie sicher, dass Sie die korrekten Werte für die Parameter VELMAX (1), ENCODER (2) und RAMPMIN (31) festgelegt haben. Falls Sie diese Einstellungen später einmal ändern, kann es sein, dass Sie die Steuerung erneut optimieren müssen.
2. Setzen Sie POSERR (15) sehr hoch, zum Beispiel auf 1000000, um zu verhindern, dass während der folgenden Testfahrten der Fehler 8 auftritt.

!!!

Sie dürfen POSERR aber nur so hoch setzen, wie es Ihre Anlage erlaubt (z.B. frei drehender Motor), denn bei diesen hohen Werten ist die Schleppfehlerüberwachung außer Kraft gesetzt.

3. Optimieren Sie die Regelung des Geschwindigkeits-Feedforwards:  
Führen Sie eine → **Testfahrt** mit den Parametern  $KPROP=0$ ,  $KDER=0$ ,  $KINT=0$ ,  $FFACC=0$  und  $FFVEL=100$  durch.  
Beurteilen Sie das Geschwindigkeitsprofil: Wenn die aktuelle Geschwindigkeitskurve niedriger ist als die Sollkurve, erhöhen Sie den Parameter  $FFVEL$ .  
Wenn die aktuelle Geschwindigkeitskurve aber höher als die geforderte Sollkurve ist, verringern Sie natürlich  $FFVEL$ .  
Führen Sie sukzessive Testfahrten durch, bis die zwei Geschwindigkeitskurven in der Grafik den gleichen Maximalwert aufweisen.  
 $FFVEL$  ist nun optimiert; sichern Sie diesen aktuellen Wert.
4. In Systemen mit einem großen Trägheitsmoment und/oder schnellen Änderungen der Sollgeschwindigkeit ist es sinnvoll, den **Beschleunigungs-Feedforward** zu nutzen (stellen Sie sicher, dass die Last angeschlossen ist, wenn Sie diesen Parameter optimieren):  
Führen Sie eine **Testfahrt** mit  $KPROP=0$ ,  $KDER=0$ ,  $KINT=0$ ,  $FFACC=0$  und  $FFVEL$  mit dem bereits optimierten Wert durch. Verwenden Sie die höchstmögliche Beschleunigung. Falls  $RAMPMIN$  (31) korrekt eingestellt ist, sollte für die Beschleunigung und das Bremsen jeweils ein Wert von 100 ausreichend sein. Beginnen sie mit einem niedrigen Wert für  $FFACC$ , etwa 10.  
Beurteilen Sie die Geschwindigkeitsgrafik: Wenn während der Beschleunigung die aktuelle Geschwindigkeit konstant geringer als die Sollgeschwindigkeit ist, geben Sie einen höheren Wert für  $FFACC$  ein und wiederholen Sie die **Testfahrt**.  
Führen Sie sukzessive Testfahrten durch, bis die zwei Geschwindigkeitskurven in der Grafik die gleichen Beschleunigungs- und Bremsrampen aufweisen.  
 $FFACC$  ist nun optimiert, sichern Sie den aktuellen Wert.
5. Als Nächstes muss der maximal mögliche stabile Wert für den **Proportionalfaktor** des PID-Reglers gefunden werden:  
Führen Sie eine **Testfahrt** mit den Parametern  $KPROP=0$ ,  $KDER=0$ ,  $KINT=0$  durch. Benutzen Sie für  $FFVEL$  und  $FFACC$  die bereits gefundenen optimierten Werte.  
Betrachten Sie die Geschwindigkeitskurve. Wenn sie nicht schwingt, erhöhen Sie  $KPROP$  und wiederholen die Testfahrt.  
Führen Sie sukzessive Testfahrten durch, bis die aktuelle Geschwindigkeitskurve leicht schwingt.  
Verringern Sie diesen ‚leicht instabilen‘ Wert von  $KPROP$  auf etwa 70 %. Sichern Sie diesen neuen Wert.
6. Um die Schwingungen, die durch den  $KPROP$  erzeugt werden, zu dämpfen, sollten Sie nun den **Differentialfaktor** optimieren.  
**Starten** Sie eine **Testfahrt** mit  $KINT=0$  und  $KDER=200$ . Setzen Sie  $FFVEL$ ,  $FFACC$  und  $KPROP$  auf die bereits gefundenen optimierten Werte.  
Führen Sie sukzessive Testfahrten mit ansteigenden **Differentialfaktoren**  $KDER$  durch. Zuerst werden die Schwingungen schrittweise geringer. Sobald die Schwingungen beginnen stärker zu werden, hören Sie auf  $KDER$  zu erhöhen.  
Sichern Sie den letzten Wert von  $KDER$ .
7. In jedem System, in dem eine statische Positionsabweichung zu Null ausgeregelt werden soll, muss der **Integralfaktor** benutzt werden. Um diesen Parameter zu setzen, müssen Sie zwischen dem schnellen Erreichen der geforderten Regelung zu Null der statischen Positionsabweichung (was gut ist) und dem Anwachsen des Überschwingens und der Schwingungen im System (was schlecht ist) abwägen.

8. Wenn Sie den **Integalfaktor** der PID-Regelung benutzen, vergessen Sie nicht KILIM so stark wie möglich zu reduzieren (natürlich ohne den KINT-Effekt zu verlieren), um die Schwingungen und das Überschwingen so gut wie möglich zu verringern.
9. Verringern Sie BANDWIDTH so stark möglich. Mit einem korrekten optimierten offenen Regelkreis kann BANDWIDTH auf mindestens 6 oder 12 % (60 – 120) verringert werden.
10. Setzen Sie POSERR (15) wieder auf einen normalen Wert von zum Beispiel 20000.

Wenn Sie die Testfahrt abgeschlossen haben → **Sichern** Sie die neuen Parameter als Benutzerparameter. Damit werden sie in der Steuerung gespeichert und künftig für alle Programme genutzt.

### Was tun, wenn ...?

- |  |   |
|--|---|
| ... Neigung zur Instabilität besteht?                | Bei einer verstärkten Neigung zur Instabilität reduzieren Sie den <b>Proportionalfaktor</b> KPROP (11) und den <b>Differentialfaktor</b> KDER (12) wieder oder setzen den <b>Integalfaktor</b> KINT (13) zurück.  |
| ... stationäre Genauigkeit fehlt?                    | Wenn eine stationäre Genauigkeit gefordert wird, müssen Sie den Integralanteil KINT (13) erhöhen.   |
| ... der Schleppabstand überschritten wird?           | Wenn die Testfahrt ständig durch die Meldung „Schleppabstand überschritten“ unterbrochen wird, setzen Sie den Parameter für den <b>Tolerierten Positionsfehler</b> POSERR (15) – innerhalb der tolerierbaren Grenze – so hoch wie möglich.<br><br>Wenn die Schleppfehler während der Beschleunigungsphase auftreten, deutet das darauf hin, dass bei den vorhandenen Lastverhältnissen die eingestellte Beschleunigung nicht erreicht werden kann. Erhöhen Sie den <b>Tolerierten Positionsfehler</b> POSERR oder ermitteln Sie eine dem Gesamtsystem angepasste maximal Beschleunigung.<br><br>Tretten Schleppfehler erst nach der Beschleunigungsphase auf und lassen sich auch durch die Erhöhung des Parameters <b>Tolerierter Positionsfehler</b> nur verzögern aber nicht eliminieren, deutet das darauf hin, dass eine zu hohe Maximaldrehzahl [U/Min] gewählt wurde. Ermitteln Sie eine dem Gesamtsystem angepasste <b>Maximalgeschwindigkeit</b> . |
| ... die maximale Beschleunigung nicht erreicht wird? | Die technischen Daten des Antriebs sind im Allgemeinen nur für ein frei drehendes Wellenende gültig. Bei einem belasteten Antrieb verringert sich die maximale Beschleunigung.<br><br>Die theoretische maximale Beschleunigung wird auch dann nicht erreicht, wenn zum Beispiel die PID-Regelung zu gering ist oder der Verstärker/Motor nicht passt und daher nicht genügend Spitzenstrom während der Beschleunigung liefert.  |

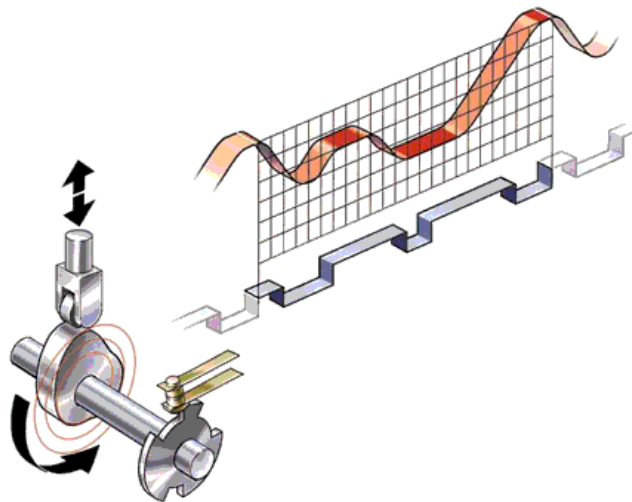


### Kurvenscheiben-Steuerung (CAM)

Um Kurvenscheibensteuerungen zu realisieren, benötigen Sie je nach Anwendung mindestens eine Kurve, die die Slave-Position in Abhängigkeit von der Master-Position sowie das Ein- und Auskuppelverhalten beschreibt. Natürlich sind für eine Kurvenscheibensteuerung weit mehr Parameter erforderlich, die zusammen mit den Fixpunkten der Kurve ein Kurvenprofil ergeben.

Die Synchronisation im CAM-Mode (Befehl SYNCC können Sie auch mit Markerkorrektur durchführen (SYNCCMM und SYNCCMS). Dies wäre zum Beispiel erforderlich, wenn die Produkte unregelmäßig auf einem Band transportiert werden oder wenn addierende Fehler ausgeglichen werden müssen.

Prinzipskizze: Links die mechanische Kurvenscheibe und die mechanische Nockenwelle, rechts die Kurven für die elektronische Kurvenscheibensteuerung und das elektronische Nockenschaltwerk:



Für die Erstellung des Kurvenprofils nutzen Sie den → **CAM-Editor**, in den Sie zuerst die bereits eingestellten Parameter laden. Dann setzen Sie die Fixpunkte der Kurve und definieren die für Ihre Anwendung erforderlichen Parameter. Alle Werte können Sie in physikalischen oder benutzerdefinierten Einheiten unter einer Windows-Oberfläche eingeben. Das Kurvenprofil können Sie ständig grafisch kontrollieren und so Geschwindigkeit und Beschleunigung der Slave-Achse prüfen.

**Interpolation** Der CAM-Editor berechnet aus den Fixpunkten die Kurve mit Hilfe einer Spline-Interpolation. Diese ist für ein minimales Drehmoment optimiert. Um Drehzahlsprünge bei mehrmaligem Kurvendurchlauf zu verhindern, wird die Geschwindigkeit am Anfang und Ende gleichgesetzt. Für diese Berechnung können Sie in der Registerkarte → **Kurwendaten** zwischen mehreren Kurventypen wählen. In jedem Fall berücksichtigt die Interpolation die Steigung der Kurve am Anfang und Ende: Entweder wird die Steigung am Anfang und Ende gemittelt, oder die Steigung am Anfang der Kurve wird auch für das Ende der Kurve benutzt, oder die Steigung am Anfang und Ende der Kurve wird auf 0 gesetzt.

**Tangentenpunkte für Geradestücke** Für Bereiche, in denen die Geschwindigkeit konstant und die Beschleunigung 0 sein muss, benutzen Sie Tangentenpunkte. Zwischen diesen Punkten wird statt eines Splines eine Gerade gelegt.

**Genauigkeit** Die Fixpunkte werden direkt als Interpolationspunkte übernommen, sofern dies der Intervallabstand zulässt. Der CAM-Editor führt zwischen den Interpolationspunkten eine lineare Interpolation durch. Wird durch den gewählten Intervallabstand ein Fixpunkt nicht getroffen, ist der entsprechende Slave-Sollwert in der Interpolationstabelle nicht vorhanden. Wenn Sie →  **Ausrichten an Gitter** aktivieren, können Sie solche Abweichungen vermeiden.

**Interne Realisierung als Array** Intern werden die Kurvenprofile als Arrays realisiert, die Sie mit einer DIM-Anweisung und dem Befehl SETCURVE aufrufen.

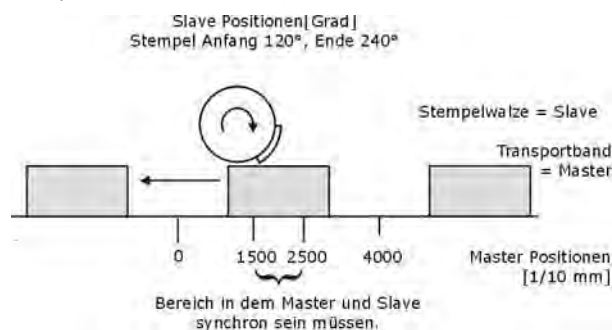
### Schnellkurs für Ungeduldige

1. CAM-Editor →  starten und Parameter mit →  **Konfiguration** aus der Steuerung lesen.  
Oder öffnen Sie eine vorhandene Konfiguration-Datei zbc mit **Datei** → **Öffnen**; dabei wird der **CAM-Editor** automatisch gestartet und die Parameter werden geladen.
2. Getriebefaktoren eingeben bzw. Benutzereinheiten MU und BE festlegen.
3. Ein neues CAM-Kurven-Array mit Sie mit  → **CAM Array hinzufügen**.  
Beachten Sie, dass **Interpolation** nicht aktiviert werden muss, außer für Steuerungen mit sehr alten Firmware Versionen (älter als 6.4.44). Falls **Interpolation aktiviert** werden muss, definieren Sie die → **Anzahl der Intervalle** für eine Masterzykluslänge. Die → **Intervall-Dauer** in der Registerkarte **Kurveninfo** sollte nicht kleiner als 20 ms sein.
4. Geben Sie mindestens drei → **Fixpunkte** für Master und Slave ein.
5. Geben Sie → **Start-Stop-Punkte** für das Ein- und Auskuppeln ein.
6. Definieren Sie in der Registerkarte **Kurvendaten** die → **Slave-Stop-Position**.
7. Geben Sie in der Registerkarte **Kurveninfo** die Anzahl der → **Zyklen / min Master** ein.
8. Mit Hilfe der grafischen Darstellung die Geschwindigkeit und Beschleunigung des Slaves prüfen.
9. Parameter in die Steuerung speichern mit →  **Konfiguration zur Steuerung schreiben**.  
Alternativ können Sie mit **Datei** → **sichern** die Parameter in eine Konfigurationsdatei zbc speichern, die Sie später in die Steuerung downloaden können.

### Beispiel: Kartons mit Haltbarkeitsdatum stempeln

Das folgende Beispiel zeigt Ihnen, wie Sie Schritt für Schritt die Kurve für diese Anwendung der Kurvenscheibensteuerung editieren und anschließend in Ihr Steuerungsprogramm einbinden.

Eine Walze soll auf Kartons eine 10 cm lange Aufschrift stempeln. Der Stempel entspricht einem Walzenabschnitt von 120 Grad. Pro Minute werden 60 Kartons auf dem Band transportiert. Die Kartons werden exakt in immer gleichem Abstand (z.B. durch ein mechanisches Raster) auf dem Band transportiert. Während des Bedruckens müssen Stempelwalze und Karton synchron laufen:



### Kurve für diese Synchronisation editieren

1. Steuerung mit den erforderlichen Parameter einstellen und diese Benutzerparameter mit **Parameter** → **Speichern in Datei** mit der Extension „zbc“ sichern.
2. Öffnen Sie diese zbc-Datei mit **Datei** → **Öffnen**.; dabei wird automatisch der **CAM-Editor** gestartet.
3. Ermitteln Sie den Getriebefaktor des Masters in MU-Einheiten.  
Die Eingabe soll in 1/10 mm Auflösung möglich sein.  
Der Antrieb ist mit dem Transportband mit einer Getriebeübersetzung von 25:11 verbunden; das heißt der Motor macht 25, das Zahnriemenrad 11 Umdrehungen.  
Getriebefaktor = 25/11

Inkrementalgeber direkt am Master-Antrieb; Encoder-Auflösung = 4096

Das Zahnriemenrad hat 20 Zähne/Umdrehung, 2 Zähne entsprechen 10 mm, daher entspricht 1 Umdrehung = 100 mm Transportbandvorschub bzw. 1000/10 mm; Skalierfaktor ist demnach 1000.

$$\frac{\text{Getriebefaktor} * \text{Drehgeberauflösung} * 4}{\text{Skalierfaktor}} q_c = 1 \text{ MU}$$

$$\frac{25/11 * 4096 * 4}{1000} q_c = 1 \text{ MU}$$

$$\frac{25 * 4096 * 4}{1000 * 11} q_c = \frac{2048}{55} q_c = 1 \text{ MU} \quad \begin{array}{l} \text{Syncfactor Master SYNCFACM (49)} \\ \text{Syncfactor Slave SYNCFACS (50)} \end{array}$$

Tragen Sie diese Werte in der Registerkarte → **Synchronisation** ein (die gewählten Einheiten sollten immer Ganzzahlen sein):

Sync Faktor Master (49) = 2048  
 Sync Faktor Slave (50) = 55

- Getriebefaktor des Slaves in Benutzereinheiten BE eingeben:

Getriebefaktor = 5/1

Drehgeber-Auflösung (Inkrementalgeber) = 500

Eine Umdrehung der Walze ist 360 Grad. Es soll mit einer Auflösung von 1/10 Grad gearbeitet werden; daher wird eine Walzenumdrehung in 3600 Arbeitseinheiten eingeteilt:

Skalierfaktor = 3600

$$\frac{\text{Getriebefaktor} * \text{Drehgeberauflösung} * 4}{\text{Skalierfaktor}} q_c = 1 \text{ BE}$$

$$\frac{5/1 * 500 * 4}{3600} q_c = \frac{5 * 500 * 4}{3600} q_c = \frac{25}{9} q_c = 1 \text{ BE} = \frac{\text{POSFACT\_Z (23)}}{\text{POSFACT\_N (26)}}$$

Tragen Sie diese ganzzahligen Werte mit **Parameter** → **Bearbeiten** in die Registerkarte **Gebersystem** ein:

Benutzerfaktor Zähler (23) = 25  
 Benutzerfaktor Nenner (26) = 9

- Definieren Sie → **Fixpunkte** für das Transportband (Master) und die Walze (Slave). Die Funktion → **Ausrichten an Gitter** sollte aktiviert sein.

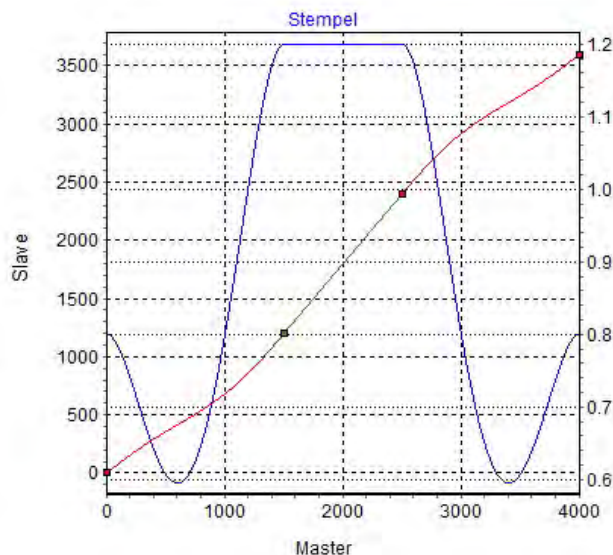
Fix Punkte		Einfügen		
Punkt	Master	Slave	Typ	
1	0	0	K	
2	1500	1200	K	
3	2500	2400	K	
4	4000	3600	K	

- Zwischen der Position 1500 und 2500 müssen Master und Slave synchron mit gleicher Geschwindigkeit fahren. Dies erfordert eine Gerade, die diese beiden Punkte verbindet.

Doppelklicken Sie in die Spalte → **Typ** in der Zeile für den Fixpunkt auf Position 1500. Wählen im darauf folgenden Dialogfenster → **Tangente** aus

Oder Sie bewegen den Cursor auf den Fixpunkt 1500, Rechtsklick und wählen im darauf folgenden Kontextmenü → **Tangente** aus. Eine Gerade (grün markiert) verbindet nun diesen Punkt und den folgenden Punkt.

7. Aktivieren Sie die grafische Darstellung der →  **Geschwindigkeit** um die entsprechende Geschwindigkeitskurve zu sehen:



8. Tragen Sie in der Registerkarte → **Kurveninfo** die → **Zyklen/min Master** = 60 ein. Das ist die Anzahl der Kartons, die (maximal) pro Minute bearbeitet werden.
9. Prüfen Sie, ob die Beschleunigung des Slaves innerhalb des Limits liegt. Aktivieren Sie dazu die Darstellung der →  **Beschleunigung** und des →  **Beschl.Limits**.
10. Um die Kurve in Ihre Steuerung zu laden, müssen Sie zuerst die Datei als zbc-Datei speichern; klicken Sie dazu auf → **Sichern als**.  
In der Titelleiste sehen Sie den Namen der Kurve und die Anzahl der Array-Elemente. Letzteres benötigen Sie für die DIM-Anweisung bei der Programmierung.
11. Laden Sie die zbc-Datei mit den veränderten Parametern und den – automatisch erzeugten – Kurvenarrays mit **Parameter** → **Wiederherstellen aus Datei** in die Steuerung.

**Programmbeispiel:** Da die Kurve intern als Array gespeichert wird, muss im Programm als erstes die DIM-Anweisung stehen:  
**Kartons mit Datum**

**stempeln**

```

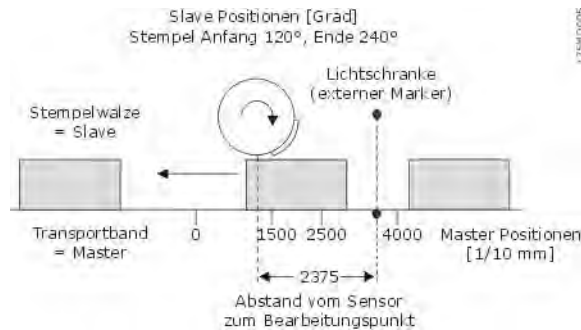
DIM stempel[92] // Anzahl der Elemente aus Titelleiste des CAM-Editors
HOME // Slave Achse führt eine Homefahrt durch (Schalter für Nullstellung oben)
// Danach befindet sich der Slave in der Nullposition (0 Grad)
// (entfällt, falls ein Absolutgeber eingesetzt wird)
SETCURVE stempel // Stempelkurve setzen
// angenommen ein Karton steht mit Vorderkante am Bearbeitungspunkt
// und der Master steht still
DEFMCPOS 1000 // 1000 entspricht dieser Position (Vorderkante Karton)
POSA CURVEPOS // Slave auf die, der Master-Position entsprechenden Kurvenposition fahren
SYNCC 0 // In den CAM-Mode wechseln und bleiben
SYNCCSTART 0 // Walze sofort mit eingestellter max. Geschwindigkeit einkuppeln
// dies verursacht keine Bewegung, da Master steht und auf korrekter Position ist
// jetzt kann der Master gestartet werden
anf: // leere Hauptschleife, damit Programm nicht beendet wird
// hier könnten weitere Verarbeitungen gemacht werden
GOTO anf
    
```

**Beispiel: Kartons bedrucken mit Markerkorrektur**

Weil die Kartons nicht in immer in exakt dem gleichem Abstand transportiert werden, benötigen Sie Marker, mit denen ein Karton erkannt und die Synchronisation korrigiert werden kann.

Im Folgenden wird beschrieben, wie Sie die Kurve des vorgehenden Beispiels für diese Anwendung anpassen.

Wieder soll eine Walze auf Kartons eine 10 cm lange Aufschrift stempeln. Auf dem Band werden pro Minute maximal 60 Kartons transportiert. Während des Bedruckens müssen Stempelwalze und Karton synchron laufen.



**Kurve für die Synchronisation mit Marker editieren**

- Schritte 1 bis 8 wie im vorhergehenden Beispiel.
- Definieren Sie in der Liste der → **Start-Stop-Punkte** die Punktepaare für das Ein- und Auskuppeln. Am Anfang des Kartons soll eingekuppelt und bis zum Ende des Kartons ausgekuppelt werden.

Punkt	Start	Stop
1	1000	1500
2	2500	3000

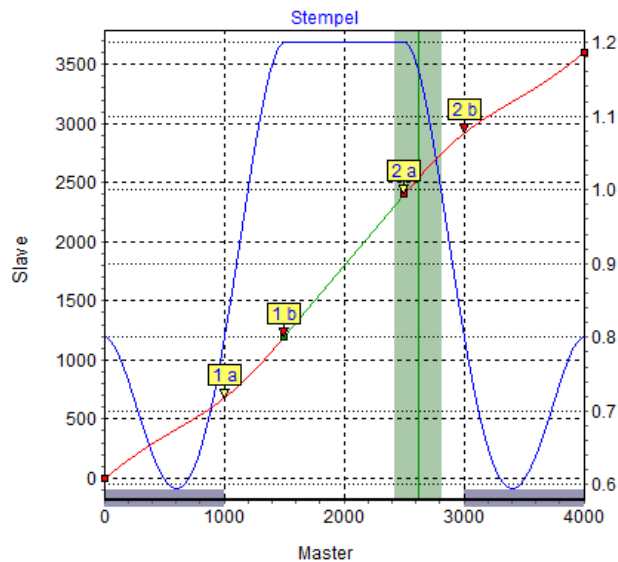
- Bestimmen Sie in der Registerkarte → **Kurvendaten** die Position, in der die Walze stoppen soll, wenn im Programm keine andere Slave-Stop-Position definiert wird:  
Die Walze soll immer auf Position 0 Grad zurückfahren: → **Slave-Stop-Position = 0**
- Die Lichtschranke (externer Marker) ist 237,5 mm vom Bearbeitungspunkt (= Stempel berührt den Karton) entfernt und erkennt den Anfang des Kartons (entspricht Master-Position 1000). Der Markerabstand beträgt demnach 2375. Tragen Sie diesen Wert in die Registerkarte → **Sync Marker** ein und definieren Sie die erlaubte Toleranz für das Auftreten der Marker und den externen Markertyp = 2 für den Master:

SYNCPULSM (58) **Markerabstand Master** = 2375  
 SYNCMWINM (68) **Master-Marker Toleranzfenster** = 200  
 SYNCMTYPM (60) **Markertyp Master** = 2

Tragen Sie die Master-Position in der **Registerkarte** → **Kurvendaten** ein:

Master-Marker-Position = 1000

- Für die Festlegung, wann die Korrektur der Synchronisation frühestens beginnen kann und wann sie beendet sein muss, betrachten Sie das Kurvenprofil: Die grüne senkrechte Linie zeigt, an welcher Master-Position der Marker erkannt wird, der hellgrüne Bereich zeigt das Toleranzfenster für das Auftreten des Master-Markers.



Die Korrektur darf frühestens beginnen, wenn ein Karton fertig bedruckt ist, denn jede Änderung der Geschwindigkeit während des Bedruckens würde den Karton beschädigen. Und die Korrektur muss vollständig beendet sein, wenn der nächste Karton den Bearbeitungspunkt erreicht. In diesem Beispiel sind die Master-Positionen Ende und Anfang eines Kartons gut geeignet:

Korrektur Start = 3000

Korrektur Ende = 1000

Tragen Sie die Werte in die Registerkarte → **Kurvendaten** ein; der Bereich wird im Kurvenprofil blau schraffiert gezeigt.

13. Prüfen Sie, ob die Geschwindigkeit und Beschleunigung des Slaves innerhalb des Limits bleiben. Aktivieren Sie dazu die Darstellung der →  **Geschwindigkeit** und des →  **Geschw.Limits** und danach die Darstellung der →  **Beschleunigung** und des →  **Beschl.Limits**.
14. Klicken Sie auf → **Speichern als** um die Datei zu speichern, zum Beispiel „marker“.
15. Laden Sie die zbc-Datei mit den veränderten Parametern und den – automatisch erzeugten – Kurvenarrays mit **Parameter** → **Wiederherstellen aus Datei** in die Steuerung.

**Programmbeispiel:**  
Kartons bedrucken mit  
Markerkorrektur

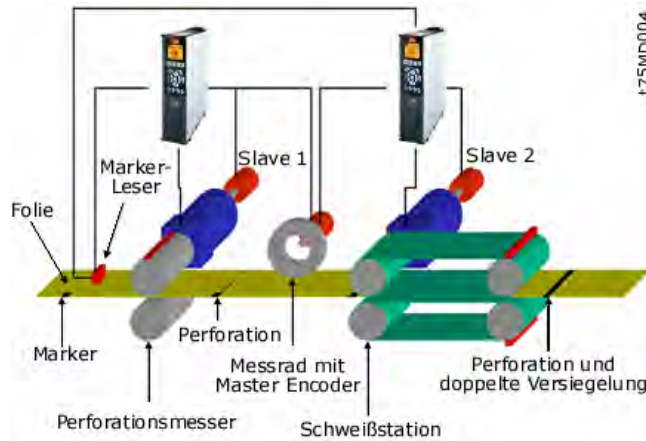
Da die Kurve intern als Array gespeichert wird, muss in Ihrem Programm als erstes die DIM-Anweisung stehen:

```

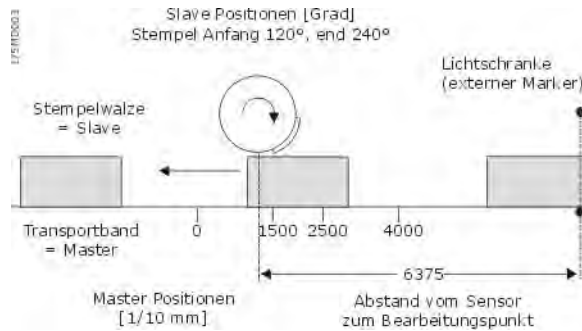
DIM marker[112] // Anzahl der Elemente aus Titelleiste des CAM-Editors
HOME // Slave Achse führt eine Homefahrt durch (Schalter für Nullstellung oben)
// Danach befindet sich der Slave in der Nullposition (0 Grad)
// (entfällt, falls ein Absolutgeber eingesetzt wird)
SETCURVE marker // Stempelkurve mit Marker setzen
dist = GET SYNCMPULSM // Abstand zum Sensor
DEFMCPOS (1000-dist) // Das ist die Stelle, die dem Sensorsignal entspricht
SET SYNCMSTART 2000 // Zählen des Masterpulses beginnt erst
// wenn nächste Flanke von Sensor kommt
SYNCCMM 0 // Im CAM-Mode synchronisieren bis Motor Stopp
SYNCCSTART 1 // Walze mit Start-Punktepaar 1 einkuppeln
// Synchronbetrieb
WAITI 4 ON // Warten auf Eingangssignal, wenn Transportband abgeschaltet wird
SYNCCSTOP 2 0 // Walze mit Stop-Punktepaar 1 auskuppeln und bei Position 0 Grad an
    
```

Wenn der Abstand des Sensors größer als eine Masterzykluslänge ist

Bei vielen Anwendungen kann der Marker nicht innerhalb einer Masterzykluslänge angebracht werden, z.B. bei folgender Maschine zur Produktion von Plastiktüten:



Da hier zwischen den Slaves keine Marker eingebaut werden können, gibt es in dieser Anwendung nur einen Markenleser, die Schweißstation liegt aber viel weiter als eine Masterzykluslänge entfernt. Da der Abstand des Sensors größer als eine Masterzykluslänge ist, wird ein Puffer für die Markerabweichung angelegt. Bei Erscheinen des Markers wird der Wert in den Puffer geschrieben und mit Erscheinen des nächsten Markers ausgelesen:

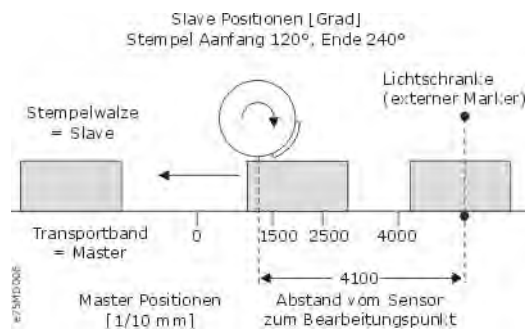


Um zu beurteilen, in welchem Bereich korrigiert werden darf, subtrahieren Sie so oft die Masterzykluslänge, bis der Wert  $< 1$  Masterzykluslänge ist. Dies ist der maximal erlaubte Abstand zum Korrigieren. In diesem Beispiel ist dieser also  $6375 - 4000 = 2375$  und damit der gleiche Korrekturbereich wie im vorangegangenen Beispiel.

## Problemfälle bei der Festlegung des Markerabstands

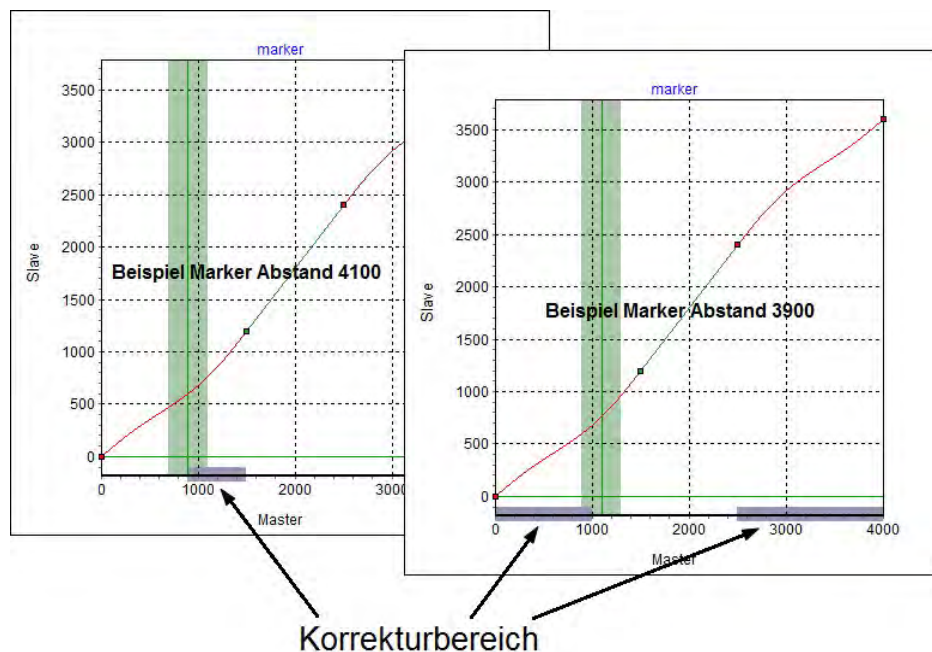
Wenn der Marker so nah am Bearbeitungspunkt angebracht ist, dass nach Erkennen des Markers keine Zeit bleibt, die Synchronisation zu korrigieren, können Sie das Problem nur durch mechanische Veränderung des Markers beheben.

Der gleiche Effekt könnte aber auch auftreten, wenn der Markerabstand größer als die Masterzykluslänge ist und nach Subtraktion dieses Wertes ebenfalls ein zu geringer Abstand bleibt, zum Beispiel:



Bei Erscheinen des Markers wird der Wert in den Puffer geschrieben. Erst wenn der nächste Marker erkannt wird, wird der Puffer ausgelesen. Das bedeutet, dass der Marker erst bei der

Master-Position 900 „erkannt“ wird und in unserem Beispiel nur noch wenig Zeit bleibt, den Fehler zu korrigieren. Es ist der gleiche Effekt, als wäre der Sensor um den Wert (Abstand – Mastertaktlänge) bzw. (4100 – 4000), also nur 10 mm vor dem Bearbeitungspunkt montiert.



Daher wäre es besser, den Sensor so zu montieren, dass der Abstand zum Bearbeitungspunkt entweder kleiner oder wesentlich größer als eine Masterzykluslänge ist, hier zum Beispiel im Abstand von 3900. Dann kann man von 2500 bis 1000 korrigieren.

Oder man montiert den Sensor weiter weg, zum Beispiel im Abstand von 7900, dies wirkt genau so, als wäre der Sensor um Abstand – Masterzykluslänge (7900 – 4000), also 3900 vor dem Bearbeitungspunkt montiert. Genügend Zeit also, um die Synchronisation zu korrigieren.

Falls dies mechanisch nicht möglich ist, muss man die Werte etwas manipulieren, damit man die Lösung mit dem Puffer vermeiden kann. Gehen Sie folgendermaßen vor:

Subtrahieren Sie vom tatsächlichen Abstand einen Wert  $x$ , damit der Abstand  $<$  Masterzykluslänge wird, zum Beispiel  $4100 - 200 = 3900$ . Den Wert  $x$  subtrahieren Sie auch von der Master-Position, also  $1000 - 200 = 800$ .

Tragen Sie beide Werte in die Registerkarten → **Sync Marker** und → **Kurvendaten** ein:

Markerabstand Master SYNCMPULSM (58) = 3900

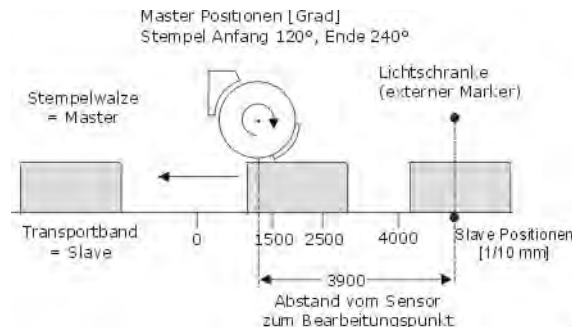
Master-Marker-Position = 800

Da nun kein Puffer erzeugt wird, könnte man zum Beispiel von 2500 bis 800 korrigieren.



### Beispiel: Slave-Synchronisation mit Marker

In folgendem Beispiel ist das Transportband der Slave und die Stempelwalze der Master, da für einen gleichmäßig Druck die Farbaufnahme und Farbabgabe kontinuierlich ablaufen müssen. Pro Minute werden maximal 20 Kartons auf dem Band transportiert. Der Abstand der Kartons ist nicht größer als eine Masterzykluslänge. Während des Bedruckens müssen Stempelwalze und Karton synchron laufen:



Im Gegensatz zur Synchronisation mit Master-Markerkorrektur wird hier die Slave-Position korrigiert und nicht die Kurve.

### Kurve für Slave-Synchronisation editieren

1. Steuerung mit den erforderlichen Parameter einstellen und diese Benutzerparameter mit **Parameter → speichern in Datei** mit der Extension „.zbc“ sichern.
2. Diese zbc-Datei mit **Datei → öffnen**. Dies startet automatisch den **CAM-Editor**.
3. Ermitteln Sie den Getriebefaktor des Masters in MU-Einheiten:

$$\text{Getriebefaktor} = 5/1$$

$$\text{Drehgeber-Auflösung (Inkrementalgeber)} = 500$$

Eine Umdrehung der Walze ist 360 Grad. Wir wollen mit einer Auflösung von 1/10 Grad arbeiten. Das bedeutet, dass eine Umdrehung der Walze in 3600 Arbeitseinheiten eingeteilt wird:

$$\text{Skalierfaktor} = 3600$$

$$\frac{\text{Getriebefaktor} * \text{Encoderauflösung} * 4}{\text{Skalierfaktor}} \text{ qc} = 1 \text{ MU}$$

Tragen Sie die Werte in der Registerkarte **→ Synchronisation** ein:

$$\text{Syncfaktor Master (49)} = 25$$

$$\text{Syncfaktor Slave (50)} = 9$$

4. Getriebefaktor des Slaves in Benutzereinheiten BE eingeben: Die Eingabe soll in 1/10 mm Auflösung möglich sein.

Der Antrieb ist mit dem Transportband mit einer Getriebeübersetzung von 25:11 verbunden; das heißt der Motor macht 25, das Zahnriemenrad 11 Umdrehungen.  
Getriebefaktor = 25/11

$$\text{Inkrementalgeber direkt am Master-Antrieb; Drehgeber-Auflösung} = 4096$$

Das Zahnriemenrad hat 20 Zähne/Umdrehung, 2 Zähne entsprechen 10 mm, daher entspricht 1 Umdrehung = 100 mm Transport.

Der Skalierfaktor ist demnach 1000.

$$\frac{\text{Getriebefaktor} * \text{Encoderauflösung} * 4}{\text{Skalierfaktor}} \text{ qc} = 1 \text{ BE}$$

Geben Sie diese Werte mit **Parameter → Bearbeiten** in die Registerkarte **Gebersystem** ein:

$$\text{Benutzerfaktor Zähler (23)} = 2048$$

$$\text{Benutzerfaktor Nenner (26)} = 55$$

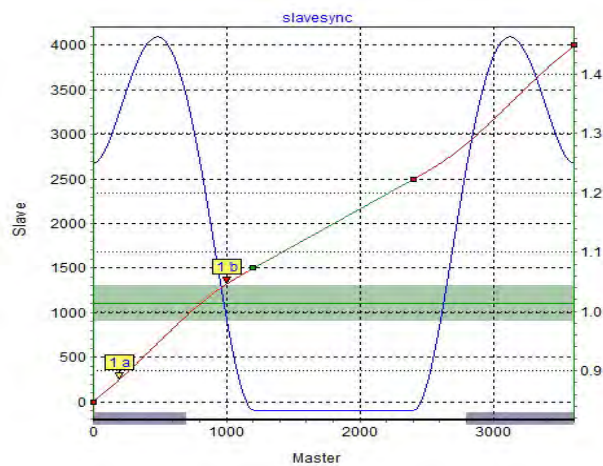
5. Definieren Sie → **Fixpunkte** für die Walze (Slave) und das Transportband (Master). Die Funktion → **Ausrichten an Gitter** sollte aktiviert sein.

Punkte	Master	Slave	Typ
1	0	0	K
2	1200	1500	K
3	2400	2500	K
4	3600	4000	K

6. Zwischen den Master-Positionen 1200 bis 2400 müssen Master und Slave synchron mit gleicher Geschwindigkeit fahren. Dafür benötigen Sie eine Gerade, die mit zwei Tangentenpunkten bestimmt wird. Mit einem Doppelklick in der Spalte → **Typ** definieren Sie für den Fixpunkt links auf Position 1200 einen Tangentenpunkt.

Punkte	Master	Slave	Typ
1	0	0	K
2	1200	1500	T
3	2400	2500	K
4	3600	4000	K

Aktivieren Sie die grafische Darstellung der →  **Geschwindigkeit** um den Verlauf zu sehen.



8. Tragen Sie in der Registerkarte → **Kurveninfo** die → **Zyklen/min Master** = 20 ein. Das ist die Anzahl der Kartons, die (maximal) pro Minute bearbeitet werden.
9. Prüfen Sie, ob die Beschleunigung des Slaves innerhalb des Limits liegt. Aktivieren Sie dazu die Darstellung der →  **Beschleunigung** und des →  **Beschl. Limits**.
10. Definieren Sie in der Liste → **Start-Stop-Punkte** um die Synchronisation am Anfang zu starten. Zwischen 20 und 100 Grad soll mit etwas Sicherheitsabstand eingekuppelt werden, denn bei 120 Grad muss aufsynchronisiert sein.

Start-Stop-Punkte		Einfügen
Punkt	Start	Stop
1	200	1000

11. Bestimmen Sie in der Registerkarte → **Kurvendaten** die Position, in der das Transportband stoppen soll, wenn im Programm keine andere **Slave-Stop-Position** definiert wird:

Das Transportband soll immer auf Position 0 halten: → **Slave-Stop-Position** = 0

12. Die Lichtschranke (externer Marker) ist 390 mm vom Bearbeitungspunkt (= Stempel berührt den Karton) entfernt und erkennt den Anfang des Kartons (entspricht Slave-

Position 1000). Der Markerabstand beträgt demnach 3900. Tragen Sie diesen Wert in die Registerkarte → **Synchronisation** ein und definieren Sie die erlaubte Toleranz für das Auftreten der Marker und den externen Markertyp = 2 für den Slave:

SYNCPULSS (59) **Markerabstand Slave** = 3900  
 SYNCMWINS (69) **Slave-Marker Toleranzfenster** = 200  
 SYNCMTYPS (61) **Markertyp Slave** = 2

Tragen Sie die Slave-Position in der Registerkarte → **Kurvendaten** ein:

Slave Marker-Position = 1000

13. Für die Festlegung, wann die Korrektur der Synchronisation frühestens beginnen kann und wann sie beendet sein muss, betrachten Sie das Kurvenprofil. Die grüne waagrechte Linie zeigt, an welcher Master-Position der Marker erkannt wird, der hellgrüne Bereich zeigt das Toleranzfenster für das Auftreten des Master-Markers.

Die Korrektur darf frühestens beginnen, wenn ein Karton fertig bedruckt ist, denn jede Änderung der Geschwindigkeit während des Bedruckens könnte den Druckstempel und/oder den Karton beschädigen. Außerdem muss die Korrektur vollständig beendet sein, wenn der nächste Karton den Bearbeitungspunkt erreicht. In diesem Beispiel sind die Slave-Positionen Ende und Anfang eines Kartons gut geeignet. Tragen Sie die Werte in die Registerkarte → **Kurvendaten** ein:

Korrektur Start = 2800  
 Korrektur Ende = 750

14. Prüfen Sie, ob die Geschwindigkeit und Beschleunigung des Slaves innerhalb des Limits bleiben. Aktivieren Sie dazu die Darstellung der →  **Geschwindigkeit** und des →  **Geschw. Limits** und danach die Darstellung der →  **Beschleunigung** und des →  **Beschl. Limits**.
15. Klicken Sie auf den Button → **Sichern als** zum Speichern.
16. Laden Sie die .zbc-Datei mit den veränderten Parametern und den – automatisch erzeugten – Kurvenarrays mit **Parameter** → **Wiederherstellen aus Datei** in die Steuerung.

**Programmbeispiel: Slave-Synchronisation mit Marker** Um die Master-Position zu bestimmen wird ein Schalter am Master vorausgesetzt, der die Nullposition signalisiert. Um den Slave in die richtige Position zu fahren, wird dieser bis zur Lichtschranke vorwärts gefahren. Dies entspricht dem Kartonanfang = 1000. Dann fährt man den Slave um 2900 (= Markerabstand 3900–1000) weiter; damit steht der Slave mit dem Kartonanfang 1000 genau vor dem Bearbeitungspunkt, also an Slave-Position 0.

```

DIM slavesync[108] // Anzahl der Elemente aus Titelleiste des CAM-Editors
HOME // Slave führt eine Homefahrt durch (Schalter für Nullstellung oben)
// Danach befindet sich der Slave in der Nullposition (0 Grad)
// (entfällt bei einem Absolutgeber)
DEFMCPOS 0 // Kurve beginnt bei Master-Position 0
SET SYNCMSTART 2000 // Zählen des Masterpulses beginnt erst
// wenn nächste Flanke vom Sensor kommt
SETCURVE slavesync // Kurve für die Slave-Synchronisation setzen
// zum Start fahren
CSTART
CVEL 10 // langsam vorwärts fahren bis Lichtschranke kommt
oldi = IPOS // oldi = letzte Markerposition des Slaves
WHILE (oldi == IPOS) DO // Warten bis Karton erkannt
ENDWHILE
POSA (IPOS + 2900) // Karton um 2900 nach vorne fahren
SYNCCMS 0 // Im CAM-Mode synchronisieren
SYNCCSTART 1 // Mit Start-Stopp-Punktepaar 1 einkuppeln
    
```

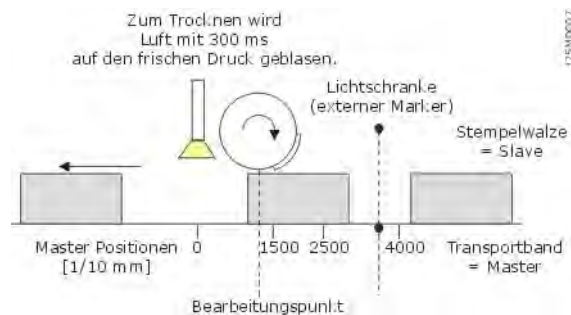
**Nockenschaltwerk** Die mechanische Nockenwelle wird ebenfalls durch eine (oder mehrere) Kurven nachgebildet. Um ein Nockenschaltwerk zu realisieren, muss es möglich sein, den Slave immer wieder an bestimmten Master-Positionen ein- und auszukuppeln.

Dies ist mit APOSS mit den Interrupt-Befehlen ON MAPOS .. GOSUB und ON APOS .. GOSUB möglich. Man kann immer dann ein Unterprogramm aufrufen, wenn eine definierte Master-Position (und zwar in positiver oder negativer Richtung) passiert wurde.

In Verbindung mit einem Kurvenprofil, in dem mehrere Start-Stopp-Punktepaare zum Aus- und Einkuppeln definiert wurden, kann man viele Anwendungen wie sie in der Verpackungsindustrie typisch sind realisieren.

### Beispiel eines Nockenschaltwerks

Nach dem Bedrucken eines Kartons soll der frische Druck sofort im Luftstrom getrocknet werden:



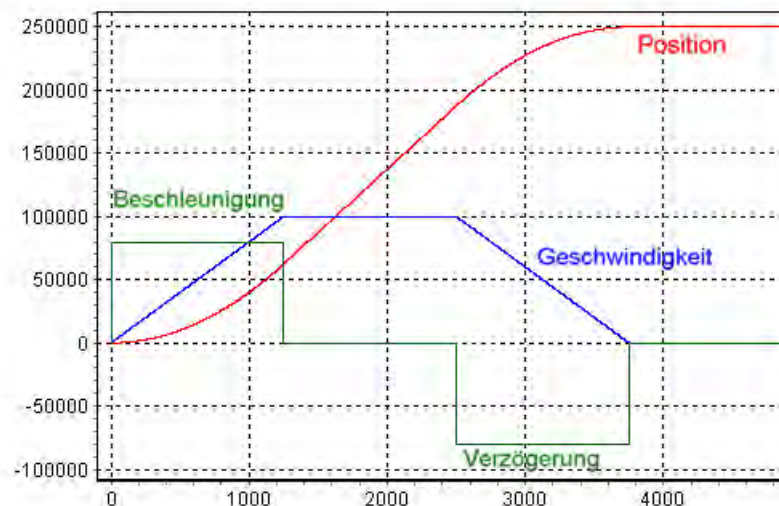
```
ON MCPOS 2500 GOSUB trocknen // Unterprogramm aufrufen, wenn die
// Master-Position 2500 in positiver Richtung passiert wurde
SUBMAINPROG
  SUBPROG trocknen
    OUT 1 1 // Trockner einschalten
    DELAY 300 // 300 ms trocknen
    OUT 1 0 // Trockner ausschalten
  RETURN
ENDPROG
```

**Ruckbegrenzung** Ruckbegrenzte Bewegungen sind ähnlich den normalen trapezförmigen Bewegungen, jedoch kann der Anwender die „Sanftheit“ der Beschleunigung und Verzögerung steuern. Dadurch kann den Ruck, der durch eine unmittelbare Beschleunigung einer trapezförmigen Bewegung verursacht wird, begrenzt werden.

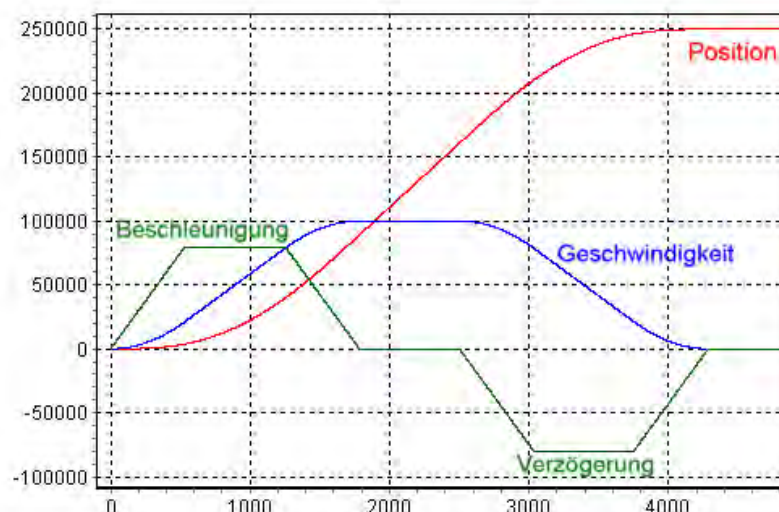
Typische Anwendungen, die ruckfreie Bewegungen erfordern, sind:

- Fahrstuhl
- Bewegung von schweren Lasten

Beispielhaft zeigt das nebenstehende Diagramm die Beschleunigungs-, Geschwindigkeits- und Positionskurve einer trapezförmigen Bewegung von einer Position zur anderen. Die scharfen Wechsel der Beschleunigung zwingen den Motor zu einem Ruck am Anfang und am Ende jeder Geschwindigkeitsrampe.



Das Diagramm zeigt die gleiche Bewegung mit einer Ruckbegrenzung. Beachten Sie, dass nun die Beschleunigung nicht mehr unmittelbar ausgeführt wird und dass die „Ecken“ der Geschwindigkeitskurve abgerundet sind. Dies resultiert in eine sanftere Motorbewegung. Es dauert außerdem etwas länger, die Zielposition zu erreichen, weil der Motor länger braucht um auf die maximale Beschleunigung zu beschleunigen.

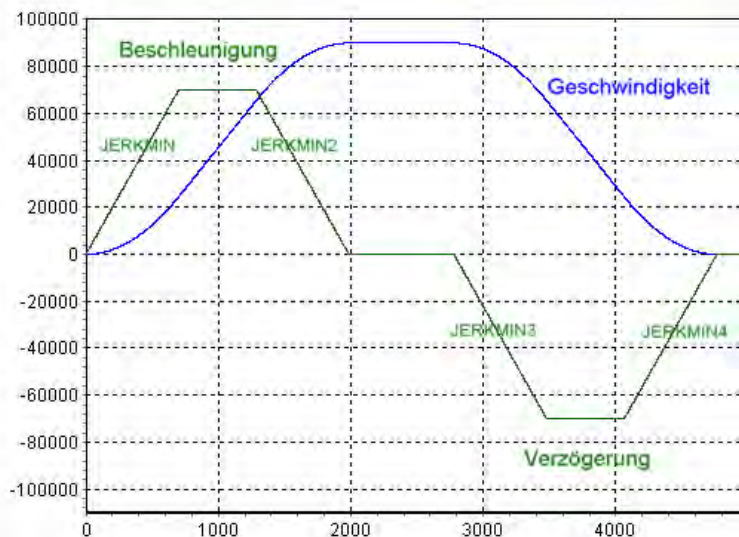


Um die „Sanftheit“ der Beschleunigungsrampe zu steuern, stehen 4 Parameter zur Verfügung:

- JERKMIN Konstante Beschleunigungsrampe beim Anfahren. Dies definiert die Zeitspanne in Millisekunden, die beim Anfahren notwendig ist, um von 0 die maximale Beschleunigung zu erreichen.
- JERKMIN2 Konstante Rücknahme der Beschleunigung. Dies definiert die Zeitspanne [ms] in der die maximale Beschleunigung auf 0 Beschleunigung reduziert werden soll (d.h. normalerweise auf konstante maximale Geschwindigkeit). Wenn „0“ gesetzt ist, wird der gleiche Wert wie bei JERKMIN verwendet.
- JERKMIN3 Konstante Verzögerungsrampe beim Anhalten. Dies definiert die Zeitspanne [ms], die notwendig ist, um von 0 die maximale Verzögerung zu erreichen. Wenn „0“ gesetzt ist, wird der gleiche Wert wie bei JERKMIN verwendet.
- JERKMIN4 Konstante Zunahme der Verzögerung. Dies definiert die Zeitspanne [ms], die notwendig ist, um von der maximalen Verzögerung auf 0 zu kommen (das ist normalerweise die Geschwindigkeit 0). Wenn „0“ gesetzt ist, wird der gleiche Wert wie bei JERKMIN verwendet.

Diese Konstanten entsprechen der „Steigung“ in den verschiedenen Teilen der Beschleunigungs-kurve (siehe folgendes Diagramm). Je größer die Werte, desto sanfter wird beschleunigt und/oder gebremst, in gleichem Masse werden die Rampen immer länger.

- !!! Die durch die *Ruckdauer* JERKMIN definierte Beschleunigungssteigung wird immer benutzt, wenn beim Anfahren beschleunigt wird und nicht nur, wenn von 0 auf die maximale Beschleunigung beschleunigt wird. Das gleiche gilt sinngemäß für die drei anderen Parameter auch: JERKMIN2 wird immer benutzt, wenn die Beschleunigung zurückgenommen wird, usw.



Ruckbegrenzte Bewegungen erreichen normalerweise nicht die Geschwindigkeits- und Beschleunigungsgrenzen, die für die Steuerung gesetzt sind (z.B. Begrenzungen durch die Befehle VEL, ACC, DEC, etc.). Im Diagramm oben sieht man diese Begrenzung an den „Plateaus“ in der Beschleunigungskurve. Falls die aktuelle Geschwindigkeit und/oder Beschleunigung außerhalb dieser Grenzen ist, wenn die ruckbegrenzte Bewegung startet, wird die Bewegung entsprechend beschleunigt oder verzögert, um sie innerhalb dieser gesetzten Grenzen zu bringen.

Es ist wichtig zu verstehen, dass für ruckbegrenzte Bewegungen die „Beschleunigung“ als „Anfahren“ in jede Richtung definiert ist (d.h. entweder vorwärts oder rückwärts) und ebenso die „Verzögerung“ als „Abbremsen“ in jede Richtung. Das Ergebnis davon ist, dass maximale Geschwindigkeit, maximale Beschleunigung, maximale Verzögerung und die vier *Ruckdauer*-Werte alle unabhängig von der Bewegungsrichtung sind. Dies kann wichtige Konsequenzen haben, wenn eine ruckbegrenzte Bewegung die Motorrichtung ändern muss, besonders wenn sich die maximale Verzögerung von der maximalen Beschleunigung unterscheidet. In diesem Fall garantiert die ruckbegrenzte Bewegung, dass die Verzögerungsrampe bei exakt Geschwindigkeit 0 sanft in eine Beschleunigungsrampe mündet, wenn die Richtung wechselt und ohne weder die Verzögerungs- noch die Beschleunigungsgrenzen zu erreichen.

Eine ruckbegrenzte Bewegung kann in drei verschiedenen Situationen benutzt werden:

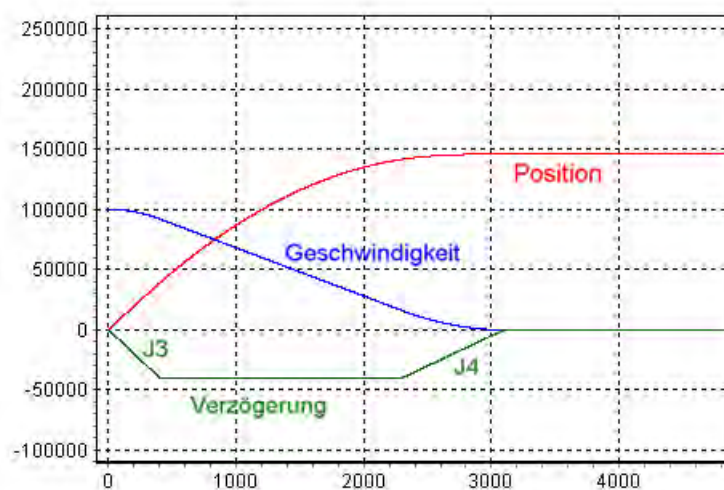
1. Anhalten aus der aktuellen Geschwindigkeit und Beschleunigung (wobei die endgültige Position nicht wichtig ist).
2. Wechsel von der aktuellen Geschwindigkeit und Beschleunigung in eine definierte konstante Geschwindigkeit (wobei die Positionen nicht wichtig sind).
3. Fahren von der aktuellen Position (und der aktuellen Geschwindigkeit und Beschleunigung) und Anhalten auf einer definierten Position.

### Beispiele für ruckbegrenzte Bewegungen

In den folgenden Beispielen ist die maximale Beschleunigung auf einen höheren Wert als die maximale Verzögerung gesetzt, so dass der Motor schneller anlaufen als bremsen kann. Ebenso ist JERKMIN kleiner gesetzt als JERKMIN2, JERKMIN2 kleiner als JERKMIN3 und JERKMIN3 kleiner als JERKMIN4, damit die verschiedenen Kurvensegmente im Diagramm besser zu unterscheiden sind. Die vier JERKMIN Werte sind mit J1, J2, J3 und J4 gekennzeichnet.

Anhalten Das folgende Diagramm zeigt eine Stopp-Bewegung, die mit einer positiven konstanten Geschwindigkeit beginnt.

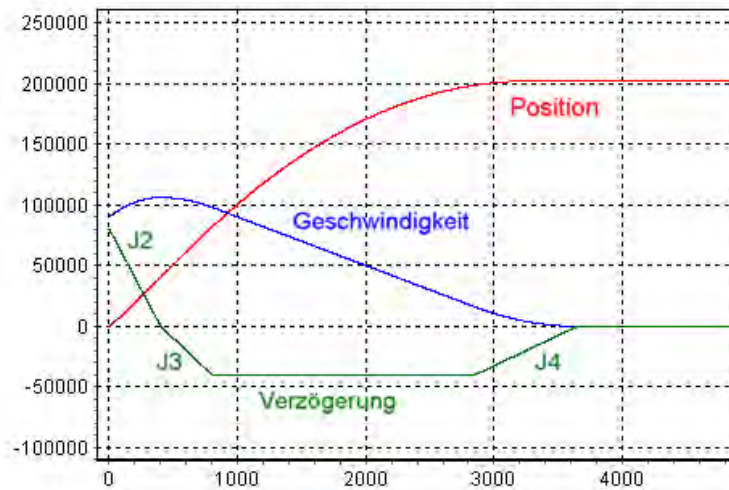
Die Kurve besteht aus einem Segment Verzögerungsrampe (JERKMIN3), gefolgt von einem Segment konstanter Verzögerung (bei maximaler Verzögerung) und schließlich einem Segment Zunahme der Verzögerung auf Geschwindigkeit 0 (JERKMIN4).



Beispiel 1

Das folgende Diagramm zeigt eine Stopp-Bewegung, die mit positiver Geschwindigkeit und positiver Beschleunigung beginnt.

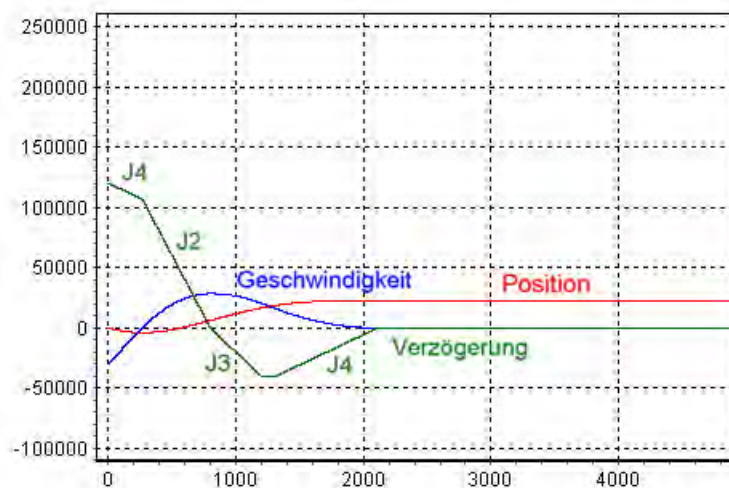
Da die anfängliche Beschleunigung positiv ist, muss die Kurve mit einer Verzögerungs-rücknahme auf Beschleunigung 0 beginnen (JERKMIN2). Es folgt dann ein Segment Verzögerungsrampe beim Anhalten (JERKMIN3), ein Segment konstante Verzögerung und ein Segment Zunahme der Verzögerung auf Geschwindigkeit 0 (JERKMIN4).



Beispiel 2

Das folgende Diagramm zeigt eine Stopp-Bewegung, die mit einer negativen Geschwindigkeit und einer sehr hohen Verzögerung beginnt. (Es ist eine Verzögerung, weil die Geschwindigkeit abnimmt.) Da jedoch die anfängliche Verzögerung so groß ist, ist der Motor nicht in der Lage ohne Überschwingen (über Geschwindigkeit 0 und zurück) zu stoppen.

Daher startet die Kurve mit einer Zunahme der Verzögerung (JERKMIN4) um die Verzögerung so stark wie möglich zu verlangsamen, bevor die Geschwindigkeit 0 erreicht wird. Bei Geschwindigkeit 0 wird aus der „Verzögerung“ eine „Beschleunigung“, weil sich die Richtung geändert hat. Demzufolge wird die Kurve mit einer Rücknahme der Beschleunigung fortgesetzt (JERKMIN2) bis Beschleunigung 0 erreicht ist. Der Motor fährt nun mit einer konstanten positiven Geschwindigkeit und daher wird die Kurve ganz normal mit einer Verzögerungsrampe (JERKMIN3), einem Segment konstanter Verzögerung (sehr kurz in diesem Beispiel) und einer Zunahme der Verzögerung auf Geschwindigkeit 0 (JERKMIN4) beendet.



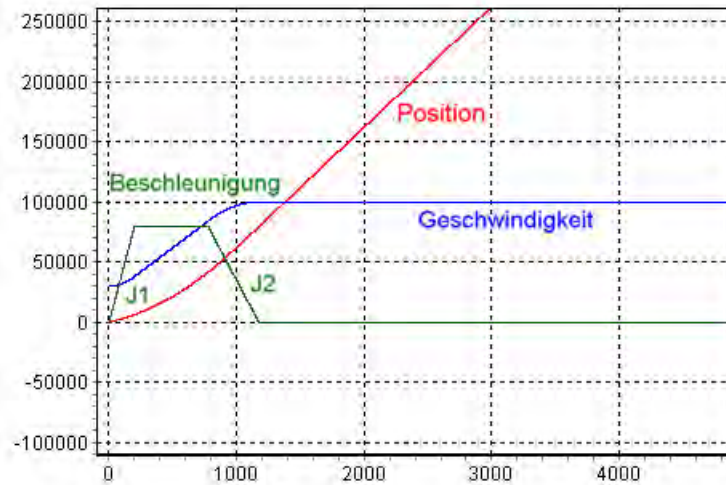
Beispiel 3



In eine konstante Geschwindigkeit wechseln

Das folgende Diagramm zeigt eine Bewegung, die mit einer positiven konstanten Geschwindigkeit beginnt und diese auf eine höhere positive konstante Geschwindigkeit steigert.

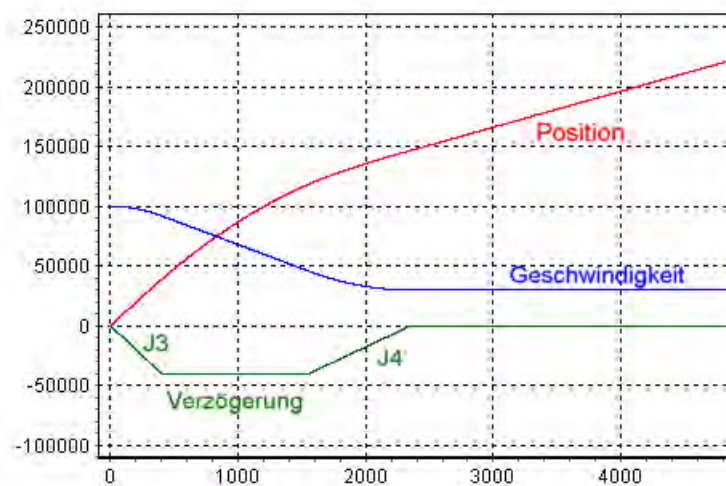
Diese Kurve besteht aus einem Segment Beschleunigungsrampe (JERKMIN), gefolgt von einem Segment konstanter Beschleunigung (bei maximaler Beschleunigung) und schließlich einer Rücknahme der Beschleunigung auf konstante Geschwindigkeit (JERKMIN2). Beachten Sie, dass die Verzögerungswerte JERKMIN3 und JERKMIN4 nicht benutzt werden, weil es nie eine Verzögerung gibt.



Beispiel 4

Das folgende Diagramm zeigt eine Bewegung, die mit einer hohen positiven konstanten Geschwindigkeit beginnt und seine Geschwindigkeit auf eine niedrigere positive konstante Geschwindigkeit verringert.

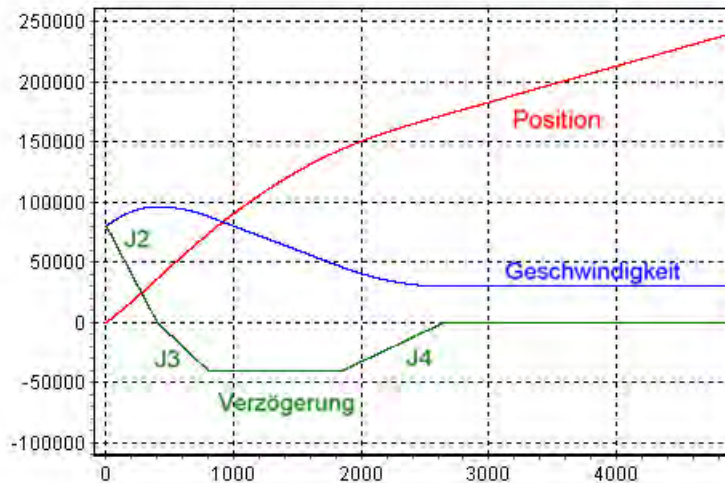
Diese Kurve besteht aus einem Segment Verzögerungsrampe (JERKMIN3), gefolgt von einem Segment konstanter Verzögerung (mit maximaler Verzögerung) und schließlich einer Zunahme der Verzögerung auf eine konstante Geschwindigkeit (JERKMIN4). Beachten Sie, dass die Beschleunigungswerte JERKMIN und JERKMIN2 nicht benutzt werden, weil es nie zu einer Beschleunigung kommt.



Beispiel 5

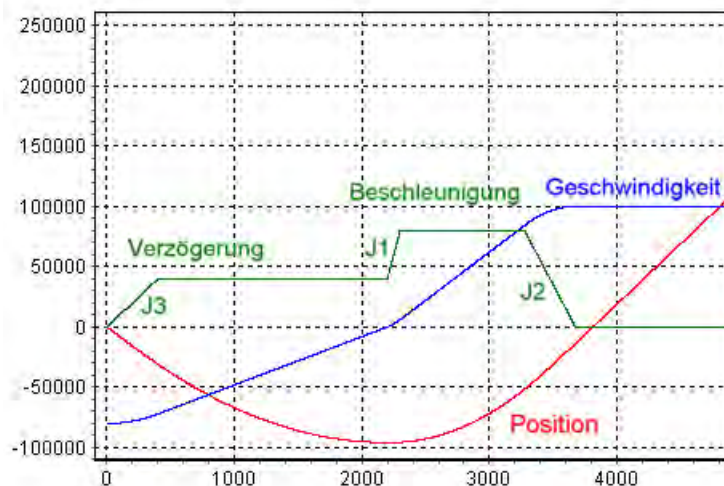
Das folgende Diagramm ist ähnlich dem vorhergehenden, außer dass es mit einer positiven Beschleunigung beginnt.

In diesem Fall muss die Kurve mit einer Rücknahme der Beschleunigung beginnen (JERKMIN2). Sobald die Beschleunigung 0 erreicht ist, wird wie im vorhergehenden Beispiel 5 fortgefahren.



Beispiel 6

Das folgende Diagramm zeigt eine Bewegung, die mit einer negativen konstanten Geschwindigkeit beginnt und dann die Richtung zu einer positiven konstanten Geschwindigkeit wechselt. Diese Kurve muss durch Abbremsen der Geschwindigkeit starten, damit sie sich „umdreht“. Daher beginnt die Kurve mit einer Verzögerungsrampe (JERKMIN3) bis sie die maximale Verzögerung erreicht. Die Verzögerung wird mit maximaler Verzögerung fortgesetzt, bis die Geschwindigkeit 0 erreicht ist. Beachten Sie, dass es kein Segment mit Verzögerungsrampe gibt, weil die Bewegung nicht anhält. Exakt bei Geschwindigkeit 0 reversiert die Richtung und die Bewegung wird nun in die andere Richtung beschleunigt. Weil aber in diesem Beispiel die maximale Beschleunigung höher ist als die maximale Verzögerung, kann ein Segment Beschleunigung eingefügt werden (JERKMIN benutzend). Die Kurve endet normal mit einem Segment konstanter Beschleunigung und einer Rücknahme der Beschleunigung auf konstante Geschwindigkeit (JERKMIN2).



Beispiel 7

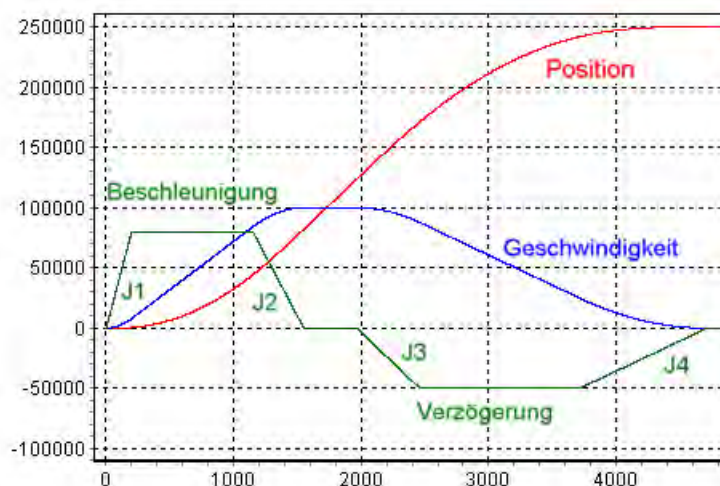
Auf eine definierte Position fahren

Das folgende Diagramm zeigt eine „normale“ Bewegung, die von einer Position, an der sie gestoppt hatte, vorwärts fährt, um an einer anderen Position anzuhalten. Die Kurve startet mit einer Verzögerung auf maximale Geschwindigkeit. Dieser Teil der Kurve ist ähnlich der ersten in „In eine konstante Geschwindigkeit wechseln“ (Beispiel 4). Dieser Kurve wechselt einfach in eine konstante Geschwindigkeit, bei der die konstante Geschwindigkeit die maximale Geschwindigkeit ist.

Daher besteht die Kurve aus einer Beschleunigungsrampe beim Anfahren (JERKMIN), einem Segment konstanter Beschleunigung mit maximaler Beschleunigung und dann einer Rücknahme der Beschleunigung auf maximale Geschwindigkeit (JERKMIN2).

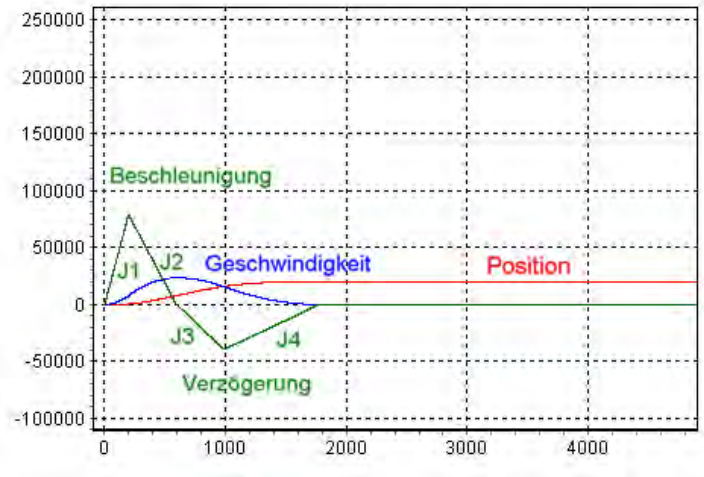
Die Bewegung wird mit maximaler Geschwindigkeit fortgesetzt, bis es notwendig wird die Verzögerungsrampe zu starten, die die Bewegung an der gewünschten Position anhält.

Die Verzögerungsrampe ist identisch zum ersten Beispiel in „Anhalten“. Die Kurve besteht aus einer Verzögerungsrampe beim Anhalten (JERKMIN3), gefolgt von einer konstanten Verzögerung (mit maximaler Verzögerung) und schließlich einer Zunahme der Verzögerung auf Geschwindigkeit 0 (JERKMIN4), um an der gewünschten Position anzuhalten.



Beispiel 8

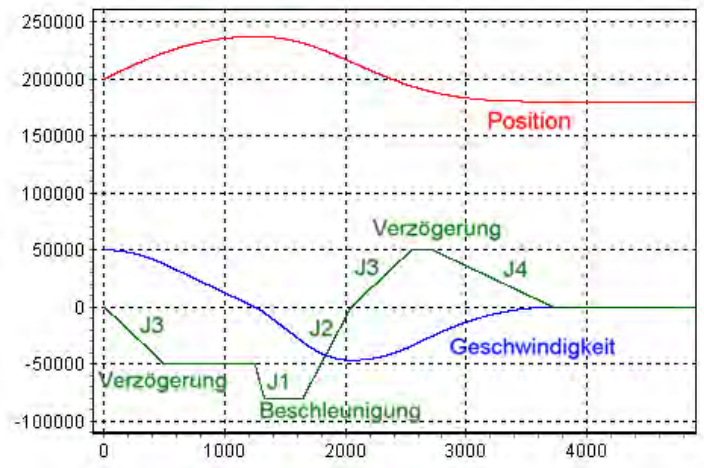
Das folgende Diagramm zeigt eine typische „kurze“ Bewegung, bei der die maximale Geschwindigkeit nicht erreicht werden kann. In diesem Fall wird so lange wie möglich mit einer Beschleunigung gefahren (JERKMIN). Abhängig davon, wie weit entfernt die Zielposition ist, kann dabei die maximale Beschleunigung erreicht werden oder nicht. An dieser Stelle wird dann die Verzögerung zurückgenommen (JERKMIN2) und sofort mit einem Segment Verzögerungsrampe fortgefahren (JERKMIN3). Abhängig von der Zielposition kann es wieder ein konstantes Verzögerungssegment geben oder nicht. Die Kurve endet mit einer Zunahme der Verzögerung bis Geschwindigkeit 0 in der Zielposition.



Beispiel 9

Das folgende Diagramm zeigt ein Beispiel bei dem der Motor anfänglich in die „falsche“ Richtung fährt, umgedreht werden und „zurück“ auf die Zielposition fahren muss. Weil er „umgedreht“ werden muss, startet die Kurve mit einer Verzögerungsrampe (JERKMIN3) bis zur maximalen Verzögerung. Dadurch wird die Geschwindigkeit verlangsamt bis der Motor umdreht. Es wird weiter mit maximaler Verzögerung abgebremst, bis die Geschwindigkeit 0 erreicht ist und die Richtung wechselt.

Exakt an diesem Punkt wird der Motor beschleunigt, aber in die andere Richtung. Von diesem Punkt an ist die Kurve gleich der einer normalen Bewegung zu einer Zielposition, außer dass die ganze Kurve invertiert wird, weil die Richtung gewechselt hat. Die Kurve hat ein Segment Beschleunigungsrampe (Rückwärtsfahrt), sie kann ein Segment konstante Beschleunigung haben oder auch nicht, sie hat ein Segment Beschleunigungsrampe, sie hat oder hat nicht ein Segment konstante Geschwindigkeit, sie hat ein Segment Verzögerungsrampe, sie hat oder hat nicht ein Segment konstante Verzögerung und sie hat schließlich eine Verzögerung zum Stoppen auf der Zielposition.



Beispiel 10