

Motion Control - Überlagerte Positionierung

Application Note 051

608 847 06_00

We automate your success.

Dieses Dokument wurde von der Jetter AG mit der gebotenen Sorgfalt und basierend auf dem ihr bekannten Stand der Technik erstellt. Änderungen und technische Weiterentwicklungen an unseren Produkten werden nicht automatisch in einem überarbeiteten Dokument zur Verfügung gestellt. Die Jetter AG übernimmt keine Haftung und Verantwortung für inhaltliche oder formale Fehler, fehlende Aktualisierungen sowie daraus eventuell entstehende Schäden oder Nachteile.

Jetter AG
Gräterstraße 2
71642 Ludwigsburg
Deutschland

www.jetter.de

Telefon:

Zentrale	+49 7141 2550-0
Vertrieb	+49 7141 2550-531
Technische Hotline	+49 7141 2550-444

E-Mail:

Technische Hotline	info@jetter.de
Vertrieb	hotline@jetter.de
	vertrieb@jetter.de

Produktname	Motion Control - Überlagerte Positionierung
Dokumentart	Application Note 051
Originaldokument	
Dokumentenversion	1.00
Artikelnummer	608 847 06_00
Ausgabedatum	22.02.2021

Inhaltsverzeichnis

1	Einleitung	1
1.1	Voraussetzungen	1
2	Überlagerte Positionierung - Einführung	2
3	Anwenden der überlagerten Positionierung	3
3.1	Befehle	3
3.1.1	<Techno>.MovePtp.StartSuperpose	3
3.1.2	<Techno>.MovePtp. StartNewTargetPositionSuperPose	3
3.1.3	<Techno>.MoveVelocity.StartSuperPose	3
3.1.4	<Techno>.MoveVelocity. ReverseSuperPose	3
3.2	Anhalten einer überlagerten Positionierung	4
3.3	Überwachen der überlagerten Positionierung	4
3.4	Berechnung der überlagerten Position	4
3.5	Ermittlung der aktuellen überlagerten Position	4
3.6	Beispiele:	5
3.6.1	Werkstückbearbeitung	5
3.6.1.1	Use Case 01: Synchronfahrt mit MovePtp.StartSuperpose und MovePtp.StartNewTargetPositionSuperPose	7
3.6.1.2	Use Case 02: Synchronfahrt mit MoveVelocity.StartSuperpose, MoveVelocity.ReverseSuperpose und MoveHalt.Start	9
3.6.1.3	Use Case 03: Synchronfahrt mit MoveVelocity.StartSuperPose und MovePtp.StartSuperPose	11
3.6.1.4	Use Case 04: Ausgekoppelt mit MovePtp.Start und MovePtp.StartSuperpose .	13
3.6.1.5	Use Case 05: Ausgekoppelt mit MovePtp.StartSuperpose / MovePtp.Start gefolgt von Einkoppeln im Modus „Sofort“, Ausgekoppelt mit gefolgt von Einkoppeln im Modus „Sofort“	14
3.6.1.6	Use Case 06: Ausgekoppelt mit MovePtp.StartSuperpose / MovePtp.Start gefolgt von Einkoppeln im Modus „Schnell“ oder „Wartend“	16
3.6.1.7	Use Case 07: Einkoppeln im Modus „Schnell“ gefolgt von MovePtp.StartSuperpose / MovePtp.Start	18
3.6.1.8	Use Case 08: MovePtp.StartSuperpose und MoveVelocity.StartSuperpose während Auskoppeln	21
3.6.1.9	Use Case 09: Auskoppeln während MovePtp.StartSuperpose und MoveVelocity.StartSuperpose	24

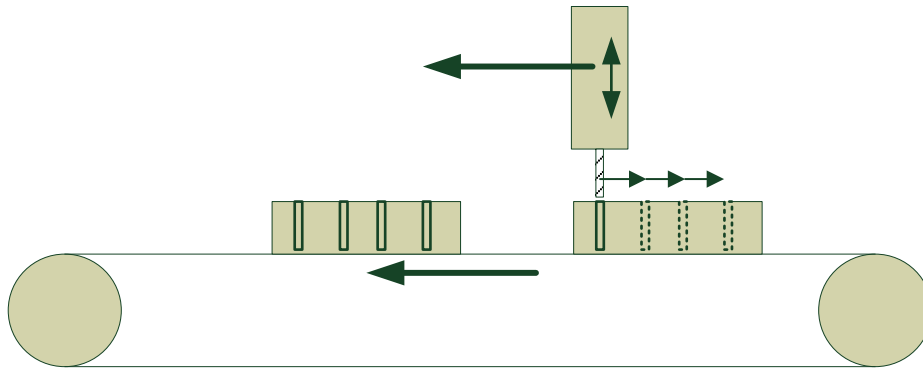
1 Einleitung

1.1 Voraussetzungen

- Für die Code- und Projektbeispiele sowie Screenshots wurden folgende Versionen verwendet:
 - JetSym 5.51
 - Motion API 1.0.0.13 bzw. Motion API 1.0.014_Beta
 - JC365MC-OS 1.30.0.00
- [Application Note 049: Motion Control - Technologieverbund](#)
- [Application Note 050: Motion Control - Anwendung eines elektrischen Getriebes im Technologieverbund](#)

2 Überlagerte Positionierung - Einführung

Eine überlagerte Positionierung ist eine Bewegung, die zusätzlich zur Grundbewegung einer Achse ausgeführt wird. Sie kann nur in Verbindung mit einem elektrischen Getriebe verwendet werden. Ist eine Folgeachse synchron zur Leitachse, folgt diese einem festen Übersetzungsverhältnis. Soll aber ein Positionsunterschied zwischen Leit- und Folgeachse im eingekoppelten Zustand verändert werden, erfolgt dies mit Hilfe der überlagerten Positionierung. Damit kann die Phasenlage der Folgeachse angepasst werden.



In diesem Beispiel soll in ein Produkt 4 Löcher gebohrt werden, während das Produkt auf einem Förderband transportiert wird. Hierzu koppelt die Bohrmaschine an der Position des ersten Bohrlochs ein. Nach der Beendigung des Bohrvorgangs wird eine überlagerte Positionierung zum nächsten Bohrloch vorgenommen. Nach der Beendigung aller Bearbeitungen koppelt die Bohrmachine wieder aus, indem auf die Grundposition gefahren wird.

Diese Application Note beschreibt die Verwendung von überlagerten Positionierungen. Dies betrifft in erster Linie diese Befehle:

```
<Techno>.MovePtp.StartSuperpose
<Techno>.MovePtp.StartNewTargetPositionSuperPose
<Techno>.MoveVelocity.StartSuperPose
<Techno>.MoveVelocity.ReverseSuperPose
```

Des Weiteren wird betrachtet, wie der Bewegungszustand der überlagerten Positionierung überwacht, eine überlagerte Positionierung gestoppt und das aktuelle Offset bestimmt wird.



INFO

Normale Positionierung

Eine nicht-überlagerte Positionierung z.B. „<Techno>.MovePtp.Start“ wird im Weiteren als „normale Positionierung“ bezeichnet.

3 Anwenden der überlagerten Positionierung

3.1 Befehle

Für die überlagerte Positionierung stehen die folgenden Befehle zu Verfügung:

<Techno>.MovePtp.StartSuperpose

<Techno>.MovePtp. StartNewTargetPositionSuperPose

<Techno>.MoveVelocity.StartSuperPose

<Techno>.MoveVelocity. ReverseSuperPose

Wie schon an "<Techno>" erkennbar wird, sind diese Befehle nur innerhalb eines Technologieverbundes verfügbar.

Im ausgekoppelten Zustand sind überlagerte Positionierungen exakt gleich wie normale Positionierungen.

Im eingekoppelten Zustand wird über die Synchronbewegung eine zusätzliche Positionierbewegung gelegt. Dabei bleibt die Folgeachse eingekoppelt. Nach Beendigung einer überlagerten Positionierung folgt die Folgeachse entsprechend ihres eingestellten Übersetzungsverhältnisses der Leitachse.

Eine normale Positionierung im eingekoppelten Zustand führt zum sofortigen Auskoppeln der Folgeachse!

Befehl	Koppelstatus	Reaktion
Move*.Start	Ausgekoppelt	Positionierung
Move*.StartSuperpose	Ausgekoppelt	Positionierung
Move*.Start	Eingekoppelt	Sofortiges Auskoppeln mit Positionierung
Move*.StartSuperpose	Eingekoppelt	Überlagerte Positionierung

3.1.1 <Techno>.MovePtp.StartSuperpose

<Techno>.MovePtp.StartSuperpose(Achse, Positioniermodus, Zielposition, Geschwindigkeit, Beschleunigung, Verzögerung, Zielfenster)

Punkt-zu-Punkt-Positionierung mit Vorgabe aller relevanten Parameter.

3.1.2 <Techno>.MovePtp. StartNewTargetPositionSuperPose

<Techno>.MovePtp. StartNewTargetPositionSuperPose (Achse, Positioniermodus, Zielposition, Zielfenster)

Punkt-zu-Punkt-Positionierung mit Vorgabe weniger Parameter, es werden die aktuell geltenden Parameter für Geschwindigkeit, Beschleunigung und Verzögerung verwendet.

3.1.3 <Techno>.MoveVelocity.StartSuperPose

<Techno>. MoveVelocity.StartSuperPose (Achse, Richtung, Geschwindigkeit, Beschleunigung, Verzögerung)

Endlospositionierung mit Vorgabe der relevanten Parameter.

3.1.4 <Techno>.MoveVelocity. ReverseSuperPose

<Techno>. MoveVelocity. ReverseSuperPose (Achse)

Richtungsänderung einer laufenden Endlospositionierung. Es werden die aktuell gültigen Parameter für Geschwindigkeit, Beschleunigung und Verzögerung verwendet. Die Richtungsänderung erfolgt mit dem für Beschleunigung gültigen Wert.

3.2 Anhalten einer überlagerten Positionierung

Soll eine überlagerte Positionierung gestoppt werden, so erfolgt dies mit dem Befehl
<Techno>.MoveHalt.Start(<Folgeachse>, Anhaltmodus, Verzögerung);

Hier gelten die gleichen Gegebenheiten, wie bei einer normalen Positionierung.

Anhaltmodus:

MCTechnoHaltModes.AtActualPosition: Es wird mit der eingestellten Verzögerung gestoppt und auf die Position zum Zeitpunkt des Kommandos zurückgefahren.

MCTechnoHaltModes.Normal: Es wird mit der eingestellten Verzögerung gestoppt.

3.3 Überwachen der überlagerten Positionierung

Für eine überlagerte Positionierung wird wie bei einer normalen Positionierung der Rampenstatus der Einzelachse abgefragt.

0: Angehalten

1: Beschleunigung

2: Konstante Geschwindigkeit

3: Verzögerung

Wird eine Folgeachse eingekoppelt oder läuft synchron zur Leitachse, dann ist der Rampenstatus der Einzelachse „Angehalten“.

Beim Starten einer Positionierung wechseln die Zustände während der einzelnen Phasen – von Beschleunigung, ggf. konstante Geschwindigkeit und Verzögerung.

Wird also zusätzlich zur Synchronbewegung eine überlagerte Positionierung ausgeführt, kann das Ende mit <Achse>.Mechanism.Slope.IsStopped abgefragt werden.



INFO

Auch wenn bei der Positionierung ein Zielfenster angegeben wird, wird im Falle einer überlagerten Positionierung der Zielfensterstatus nicht gesetzt.

3.4 Berechnung der überlagerten Position

Bei der Verwendung der überlagerten Positionierung ergibt sich für die Folgeachse eine resultierende Sollposition aus der Summe der Position aufgrund des elektrischen Getriebes und der überlagerten Positionierung.

$$x(t) = x_{EGetriebe}(t) + x_{ptp}(t)$$

Der Startwert von x_{ptp} ist beim Einkoppeln immer 0.

Wird eine überlagerte Absolutposition auf z.B. 10° vorgenommen und anschließend noch einmal, so findet beim zweiten Mal keine Bewegung statt.

3.5 Ermittlung der aktuellen überlagerten Position

Der aktuelle Wert von x_{ptp} ist nicht direkt aus der MCX auslesbar.

Deshalb müssen hierzu die aktuellen Ist-/oder Sollpositionen der Leit- und der Folgeachse ausgelesen werden. Der Unterschied aus der Position der Leit- und der Folgeachse abzüglich des Anfangsunterschieds zum Einkoppelzeitpunkt ergibt die aktuelle überlagerte Position.

$$x_{ptp}(t) = x(t) - x_{Leitachse}(t) - x_{offset}(Einkoppelzeitpunkt)$$

Ist das Übersetzungsverhältnis ungleich 1, und/oder die Modulo-Verfahrensbereiche unterschiedlich bzw. sind schon mehrere Modulozyklen gefahren worden, dann kann obige Formel nicht ausreichen.

Alternativ sollte dementsprechend die überlagerte Positionierung mitgezählt werden.

Bei überlagerten Absolutpositionierungen ist automatisch die Zielposition nach der Positionierung die überlagerte Position.

Bei überlagerten Relativpositionierungen muss die Distanz aufsummiert werden.

Bei überlagerten Endlospositionierungen empfiehlt es sich, vor dem Positionierkommando die aktuelle Sollposition auszulesen, mögliche Moduloumbrüche der Folgeachse mitzuzählen und beim Anhalten der Endlospositionierung wieder die nun aktuelle Sollposition der Folgeachse auszulesen. Die Differenz der Positionen plus die durchfahrenen Modulobereiche ergeben dann die überlagerte Position.

3.6 Beispiele:

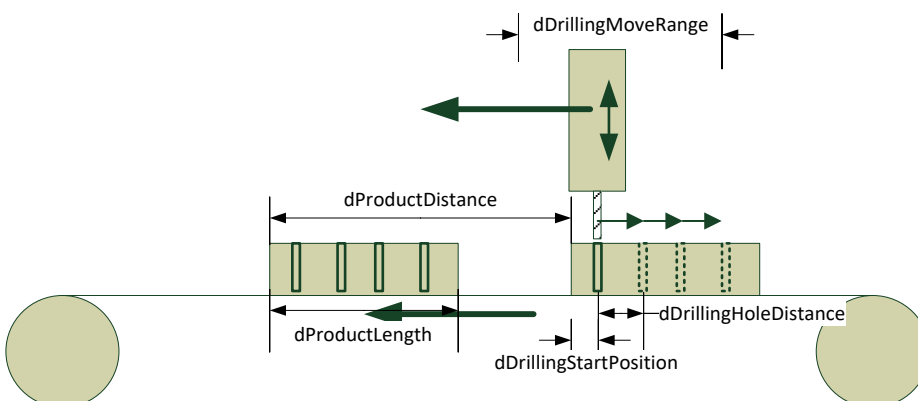
3.6.1 Werkstückbearbeitung

Siehe Beispielprojekt „[ContinuousProductDrilling](#)“

In diesem Beispiel soll in ein Produkt in einem festen Abstand Löcher gebohrt werden, während das Produkt auf einem Förderband transportiert wird. Hierzu koppelt die Bohrmaschine an der Position des ersten Bohrlochs ein. Nach der Beendigung des Bohrvorgangs wird eine überlagerte Positionierung zum nächsten Bohrloch vorgenommen. Nach der Beendigung aller Bearbeitungen koppelt die Bohrmaschine wieder aus, indem auf die Grundposition gefahren wird.

Diese Anwendung ist im Prinzip eine „Fliegende Säge“-Anwendung. Zu dieser gehört im Normalfall aber eine Druckmarkenerfassung zur Erkennung der Position des Produkts auf dem Band mit der entsprechenden Verrechnung zur Aufsynchronisierung des Werkzeugs. Zur Vereinfachung des Beispiels wird aber ein fester Produktabstand angenommen, so dass das Produkt immer an der gleichen Position erwartet wird.

Zur weiteren Vereinfachung verzichtet das Beispiel auf Überwachungen und entsprechende Fehlerreaktionen. So sollte in einer realen Anwendung überprüft werden, dass während der Bearbeitung der Verfahrensbereich des Werkzeugvorschubs nicht über-/unterschritten wird. Angenommen, der Werkzeugvorschub hat einen Verfahrensbereich von 500 mm. Die erste Bearbeitung beginnt bei Werkzeugvorschub-Position 450 mm und das Werkstück wird mit 100 mm/s transportiert. Das Bohren dauert 1 s. Innerhalb dieser Bearbeitungszeit fahren das Werkstück und somit auch der Werkzeugvorschub 100 mm. Dann wäre an der Werkzeugvorschub-Position 550 mm das Bohren beendet. Da aber bei 500 mm das Ende des Verfahrensbereichs erreicht ist, würde hier ein Fehler auftreten. Ähnliche Szenarien sind auch für die gegenüberliegende Verfahrensbereichsgrenze denkbar.



Hierbei werden folgende Parameter verwendet:

dProductDistance: Fester Produktabstand

dProductLength: Länge des Produkts

dDrillingMoveRange: Verfahrensbereich der Bohrmaschine

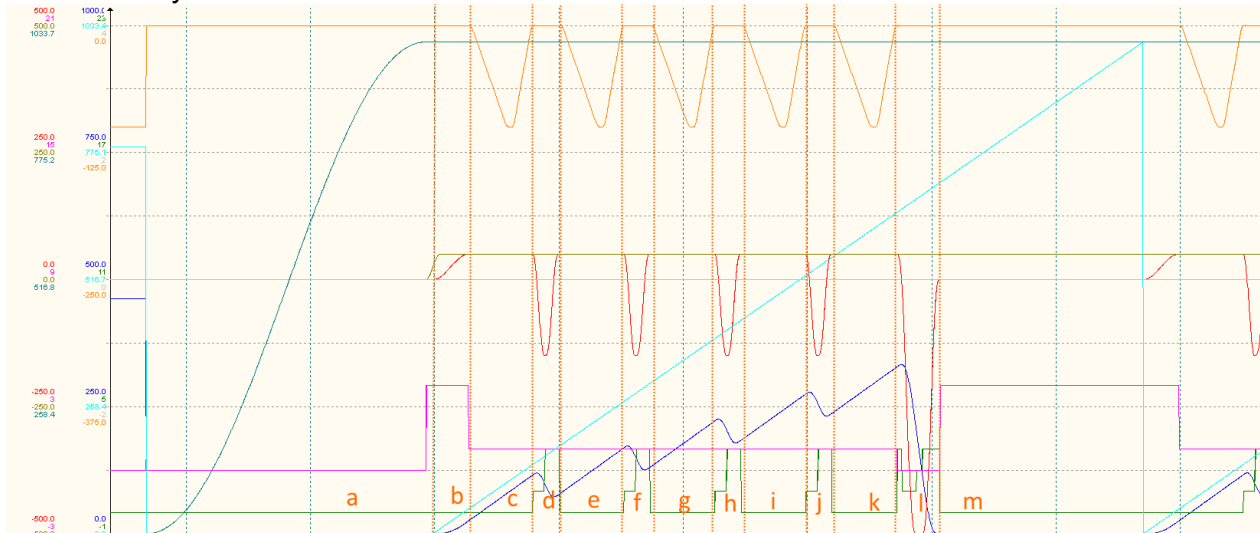
dDrillingStartPosition: Position des ersten Bohrlochs

dDrillingHoleDistance: Abstand zwischen den Bohrlöchern

Die Hardwarekonfiguration besteht aus:

- xConveyor: Linear/Modulo - Transportband, welches das Produkt transportiert
- xDrillingMove: Linear/Normal - Verfährt Bohrmaschine parallel zum Transportband
- xDrillingLift: Linear/Normal - Vertikale Bewegung der Bohrmaschine
- xDriller: Rotatorisch/Modulo - Drehbewegung der Bohrmaschine

Der Technologieverbund „tecDrilling“ besteht aus „xConveyor“ als Leitachse und „xDrillingMove“ als E-Getriebe-Folgeachse, so dass sich das Werkzeug auf das Transportband und damit auf das Produkt auf synchronisieren kann.



Legende:

Dunkelblau: Sollposition Vorschubachse (xDrillingMove)

Rot: Sollgeschwindigkeit Vorschubachse (xDrillingMove)

Pink: Koppelstatus Vorschubachse (xDrillingMove)

Grün: Rampenstatus Vorschubachse (xDrillingMove)

Hellblau: Sollposition Transportband (xConveyor)

Oliv: Sollgeschwindigkeit Transportband (xConveyor)

Grau: Sollgeschwindigkeit Bohrmaschine (xDriller)

Orange: Sollposition vertikale Bewegung der Bohrmaschine (xDrillingLift)

- Die Bohrmaschine startet und beschleunigt auf Zielgeschwindigkeit. Alle anderen Achsen stehen auf Grundposition. Die Vorschubachse ist ausgekoppelt.
- Sobald die Zielgeschwindigkeit der Bohrmaschine erreicht ist, startet das Transportband. Die Vorschubachse wird „schnell“ eingekoppelt. Der Koppelstatus wechselt auf „Einkoppeln“.
- Die Vorschubachse ist eingekoppelt und befindet sich nun über dem ersten zu bohrenden Loch. Das Bohren wird nun gestartet: Die Achse xDrillingLift taucht in das Werkstück ein und fährt auf Grundstellung zurück. Während dieser Bearbeitungszeit fährt die Vorschubachse synchron zum Transportband.
- Die Bohrung ist fertig. Mit Hilfe der überlagerten Positionierung wird zum nächsten Bohrloch gefahren. Im Oszilloskop ist zu erkennen, dass die Position der Vorschubachse kleiner wird und auch die Geschwindigkeit niedriger als Synchrongeschwindigkeit ist. Anhand des Rampenstatus läßt sich die überlagerte Positionierung überwachen.
- ... k) Es wiederholen sich die Punkte c) und d).

- l) Es sind alle Löcher gebohrt. Die Vorschubachse fährt mit einer normalen Positionierung auf die Grundposition.
Die Vorschubachse wird dadurch direkt ausgekoppelt.
- m) Nach Erreichen der Grundposition wird die Vorschubachse direkt wieder eingekoppelt, so dass der Koppelstatus auf „Einkoppeln“ wechselt.
Ein neues Produkt kommt auf dem Transportband an und wird entsprechend bearbeitet.

Projekt "MoveSuperpose"

Im Projekt „[MoveSuperpose](#)“ sind verschiedene Anwendungsfälle dargestellt:

- Use Case 01: Synchronfahrt mit MovePtp.StartSuperpose und MovePtp.StartNewTargetPositionSuperPose
- Use Case 02: Synchronfahrt mit MoveVelocity.StartSuperpose, MoveVelocity.ReverseSuperpose und MoveHalt.Start
- Use Case 03: Synchronfahrt mit MoveVelocity.StartSuperpose und MovePtp.StartSuperpose
- Use Case 04: Ausgekoppelt mit MovePtp.Start und MovePtp.StartSuperpose
- Use Case 05: Ausgekoppelt mit MovePtp.StartSuperpose / MovePtp.Start gefolgt von Einkoppeln im Modus „Sofort“
- Use Case 06: Ausgekoppelt mit MovePtp.StartSuperpose / MovePtp.Start gefolgt von Einkoppeln im Modus „Schnell“
- Use Case 07: Einkoppeln im Modus „Schnell“ gefolgt von MovePtp.StartSuperpose/MovePtp.Start
 - Vor optimalem Synchronpunkt
 - Während Einkoppelfahrt
- Use Case 08: MovePtp.StartSuperpose und MoveVelocity.StartSuperpose während Auskoppeln
- Use Case 09: Auskoppeln während MovePtp.StartSuperpose und MoveVelocity.StartSuperpose

3.6.1.1 Use Case 01: Synchronfahrt mit MovePtp.StartSuperpose und MovePtp.StartNewTargetPositionSuperPose

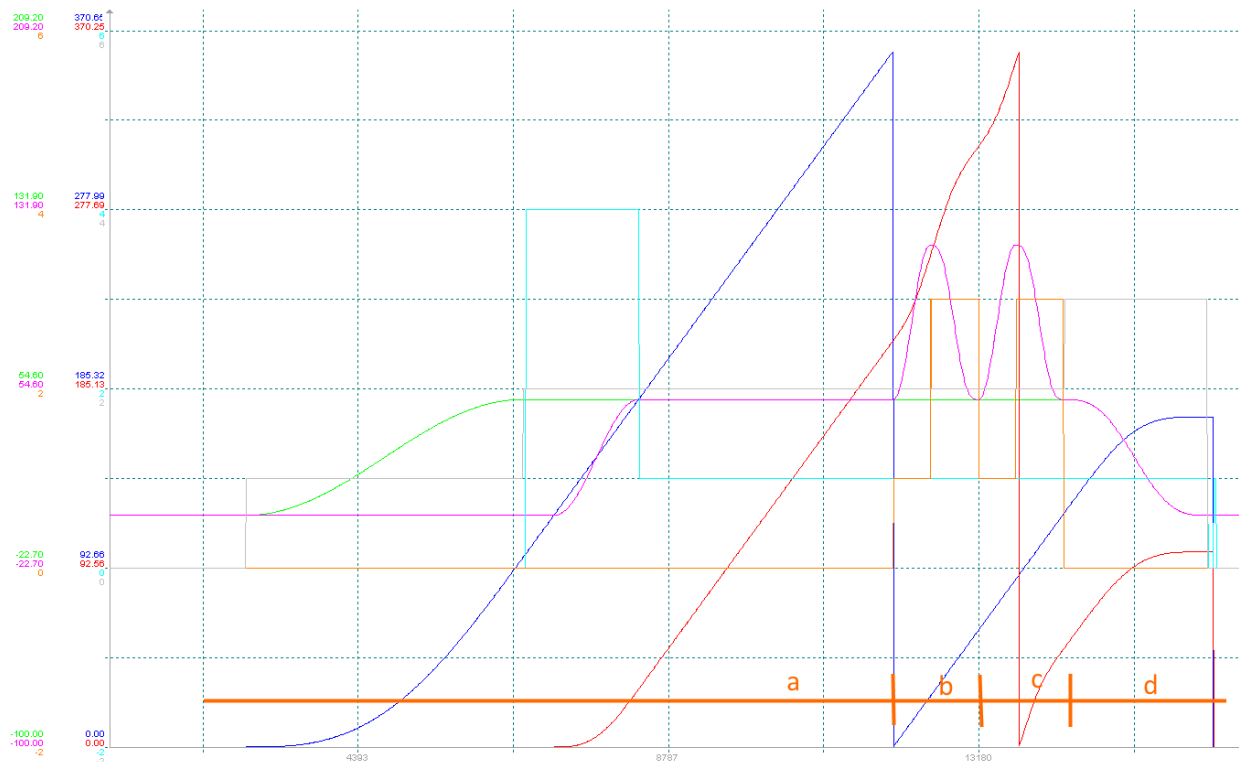
In diesem Anwendungsfall soll gezeigt werden, wie sich die beiden Befehle auswirken, wenn während der Synchronfahrt eine überlagerte Positionierung gestartet wird. Es wird ersichtlich, wie die Rampenstatus wechseln.

Die Befehle „*StartSuperPose*“ und „*StartNewTargetPositionSuperPose*“ sind im Verhalten und in der Wirkung nahezu identisch, nur dass beim ersten Befehl noch alle Positionierparameter wie Geschwindigkeit, Beschleunigung und Verzögerung angegeben werden können.

Bei beiden Befehlen kann auch ein Zielfenster angegeben werden, welches aber nicht ausgewertet wird, falls die Folgeachse gekoppelt ist. (siehe auch Use Case 04, im ausgekoppelten Zustand wirken die Befehle wie nicht überlagerte Positionierung, also „*MovePtp.Start*“ und „*MovePtp.StartNewTargetPosition*“)

```
when xvFollower.State.Techno.IsCoupled continue;
when xvMaster.Position.Setpoint < 100.0 continue;
tecGear.MovePtp.StartSuperPose(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 40.0, 100.0,
200.0, 200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;
tecGear.MovePtp.StartNewTargetPositionSuperPose(xvFollower, MCTechnoPositioningModes.AbsModuloAuto,
80.0);
when xvFollower.Mechanism.Slope.IsStopped continue;
```

Laut Code soll zu einem bestimmten Zeitpunkt eine überlagerte Positionierung „*StartSuperPose()*“ gestartet werden. Ist diese beendet – Abfrage mit „*Slope.IsStopped*“, wird die nächste überlagerte Positionierung gestartet. Auch hier wird wieder auf das Ende der Positionierung „*StartNewTargetPositionSuperPose()*“ mit „*Slope.IsStopped*“ gewartet.



Legende:

- Sollposition der Leitachse
- Sollgeschwindigkeit der Leitachse
- Sollposition der Folgeachse
- Sollgeschwindigkeit der Folgeachse
- Koppelstatus der Folgeachse(0: ausgekoppelt, 1: eingekoppelt, 3: auskoppeln, 4: einkoppeln)
- Rampenstatus der Folgeachse (0: Angehalten, 1: Beschleunigung, 2: Konstante Geschwindigkeit, 3: Verzögerung)

Phasen:

- a) Die Leitachse beschleunigt auf die Zielgeschwindigkeit. Sobald diese erreicht ist, wird die Folgeachse eingekoppelt. Der Koppelstatus wechselt auf „Einkoppeln“ gefolgt von „Eingekoppelt“. Die Folgeachse fährt synchron zur Leitachse und hat auch wie im Oszilloskop ersichtlich die gleiche Geschwindigkeit.
- b) An der Modulogrenze wird die erste überlagerte Positionierung „*StartSuperPose()*“ gestartet. Die Sollposition steigt nicht mehr linear an, sondern die Steigung wird größer. Besser ist dies erkennbar an der Sollgeschwindigkeit der Folgeachse. Diese wird kurz größer, um dann wieder bis auf Synchronengeschwindigkeit kleiner zu werden. Der Rampenstatus wechselt von „angehalten“, auf „beschleunigen“ und direkt auf „verzögern“, da für diese Positionierung die Zielgeschwindigkeit nicht erreicht wird. Mit anderen Positionierparametern wird auch diese erkennbar. Mit Erreichen der überlagerten Zielposition wechselt der Rampenstatus auf „angehalten“.
- c) Die zweite überlagerte Positionierung *StartNewTargetPositionSuperPose()* wird gestartet. Da in diesem Beispiel die Positionierparameter gleich der ersten Positionierung sind, ist der Verlauf identisch. Es wird auch um 40° weiter gefahren.
- d) Die Leitachse wird angehalten und am Ende die Folgeachse ausgekoppelt.

3.6.1.2 Use Case 02: Synchronfahrt mit MoveVelocity.StartSuperpose, MoveVelocity.ReverseSuperpose und MoveHalt.Start

In diesem Anwendungsfall soll gezeigt werden, wie sich die beiden Befehle auswirken, wenn während der Synchronfahrt eine überlagerte Endlospositionierung gestartet wird. Es wird ersichtlich, wie die Rampenstatus wechseln.

Mit MoveHalt.Start wird die überlagerte Endlospositionierung beendet, so dass im Anschluss die Folgeachse weiter mit Synchrongeschwindigkeit der Leitachse folgt.

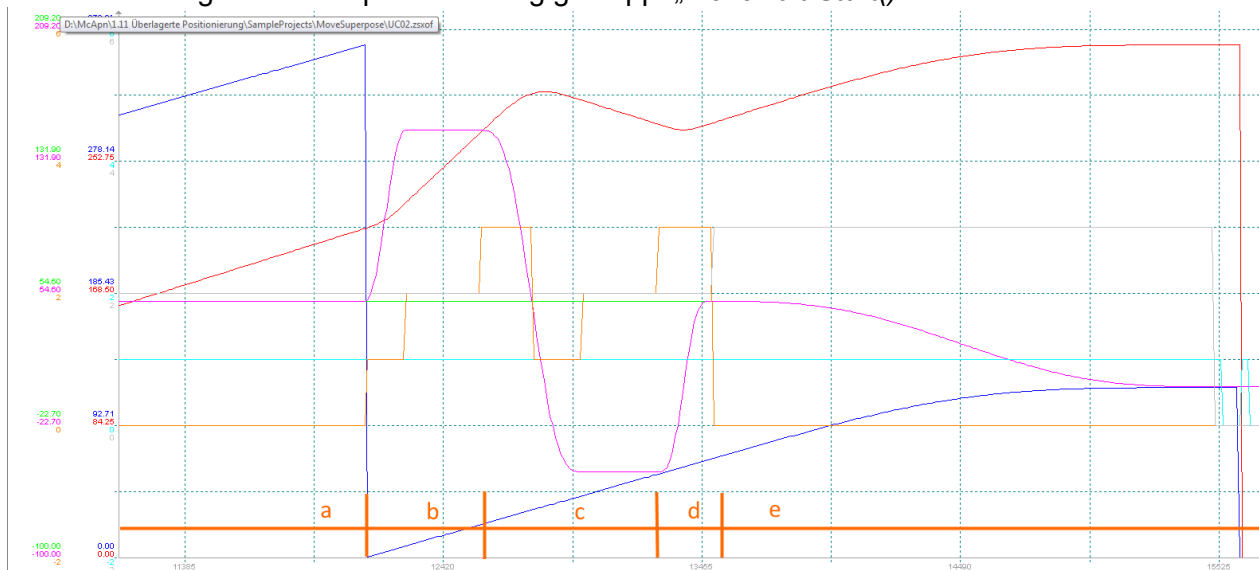
```
when xvFollower.State.Techno.IsCoupled continue;

when xvMaster.Position.Setpoint < 100.0 continue;
tecGear.MoveVelocity.StartSuperPose(xvFollower, Directions.Positive, 100.0, 1000.0, 1000.0);
when xvFollower.Mechanism.Slope.IsAtConstantSpeed continue;
delay(t#300ms);

tecGear.MoveVelocity.ReverseSuperPose(xvFollower);
when xvFollower.Mechanism.Slope.IsAtConstantSpeed continue;
delay(t#300ms);
tecGear.MoveHalt.Start(xvFollower, MCTechnoHaltModes.Normal);
when xvFollower.Mechanism.Slope.IsStopped continue;
```

Zu einem bestimmten Zeitpunkt wird die überlagerte Endlospositionierung „MoveVelocity.StartSuperPose()“ gestartet. Zu einem anderen Zeitpunkt – hier 300 ms nach Erreichen der Zielgeschwindigkeit zur besseren Darstellung im Oszilloskop – wird die überlagerte Endlospositionierung reversiert „MoveVelocity.ReverseSuperPose()“.

Auch hier werden 300 ms nach Erreichen der neuen Zielgeschwindigkeit gewartet. Anschließend wird die überlagerte Endlospositionierung gestoppt „MoveHalt.Start()“.



Legende ([siehe Legende Use Case 01](#))

Phasen:

- Die Leitachse beschleunigt auf die Zielgeschwindigkeit. Sobald diese erreicht ist, wird die Folgeachse eingekoppelt. Der Koppelstatus wechselt auf „Einkoppeln“ gefolgt von „Eingekoppelt“. Die Folgeachse fährt synchron zur Leitachse und hat auch wie im Oszilloskop ersichtlich die gleiche Geschwindigkeit.
- An der Modulogrenze wird die überlagerte Endlospositionierung „MoveVelocity.StartSuperPose()“ gestartet. Die Sollgeschwindigkeit der Folgeachse steigt von der Synchrongeschwindigkeit um die vorgegebene Zielgeschwindigkeit an. Die vorher parallel zur Sollposition der Leitachse verlaufende Sollposition der Folgeachse

wird steiler.

Der Rampenstatus wechselt von „angehalten“, „beschleunigen“ auf „konstante Geschwindigkeit“.

- c) Nach 300 ms wird die aktuell laufende überlagerte Endlospositionierung reversioniert „*MoveVelocity.ReverseSuperPose()*“.

Die Sollgeschwindigkeit der Folgeachse verringert sich. Sie verringert sich sogar soweit, dass eine Drehrichtungsumkehr der Folgeachse stattfindet.

Ab dem Zeitpunkt der Drehrichtungsumkehr sinkt die Sollposition der Folgeachse.

Am Rampenstatus ist gut zu sehen, dass beim Reversionieren erst bis auf die überlagerte Geschwindigkeit 0 verzögert wird, um dann auf die neue, jetzt negative Geschwindigkeit zu beschleunigen.

- d) Nach 300 ms wird die aktuell laufende überlagerte Endlospositionierung angehalten „*MoveHalt.Start()*“.

Die absolute Geschwindigkeit der überlagerten Endlospositionierung nimmt ab, was sich hier in einem Geschwindigkeitsanstieg auf Synchrongeschwindigkeit bemerkbar macht.

Der Rampenstatus wechselt auf „verzögern“ und weiter auf „angehalten“

Es ist jetzt auch erkennbar, dass nur die überlagerte Endlospositionierung angehalten und nicht die Achse gestoppt wird.

- e) Zum Ende des Anwendungsfalls wird die Leitachse gestoppt.

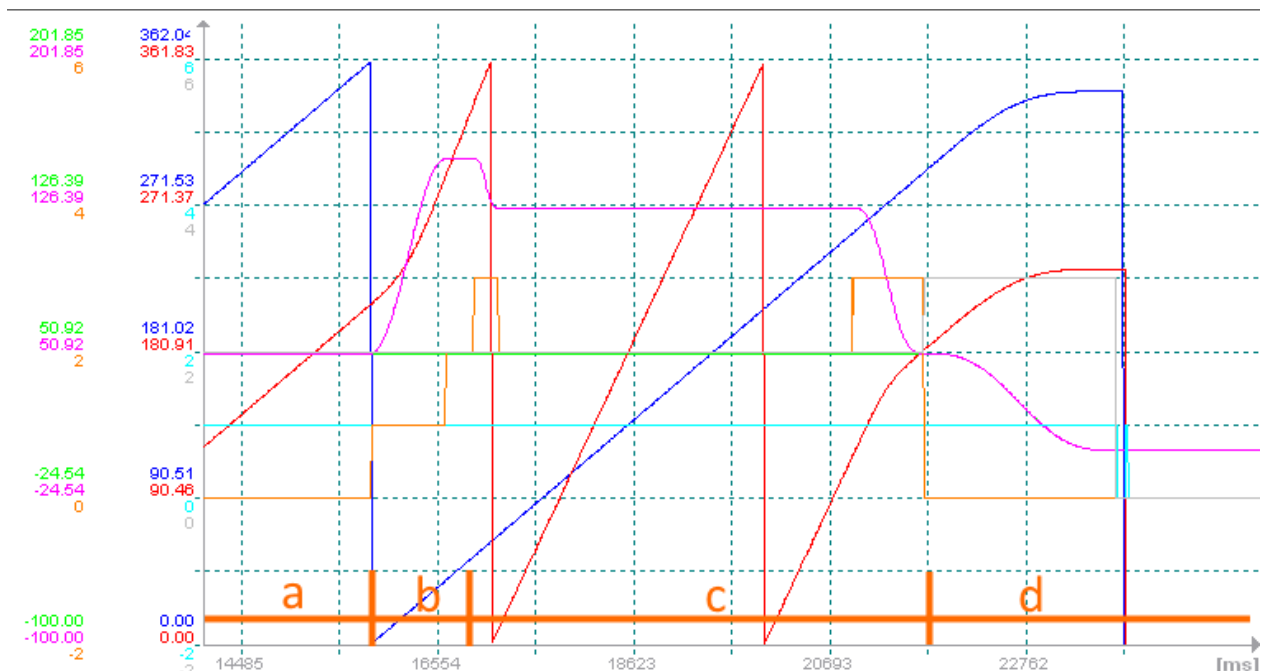
3.6.1.3 Use Case 03: Synchronfahrt mit MoveVelocity.StartSuperPose und MovePtp.StartSuperPose

In diesem Anwendungsfall soll prinzipiell gezeigt werden, wie sich die Befehle auswirken, wenn während der Synchronfahrt eine überlagerte Endlospositionierung gefolgt von einer überlagerten Positionierung gestartet wird. Es wird ersichtlich, wie die Rampenstatus wechseln. Mit MoveHalt.Start wird die überlagerte Endlospositionierung beendet, so dass im Anschluss die Folgeachse weiter mit Synchrongeschwindigkeit der Leitachse folgt.

```
when xvFollower.State.Techno.IsCoupled continue;

when xvMaster.Position.Setpoint < 100.0 continue;
tecGear.MoveVelocity.StartSuperPose(xvFollower, Directions.Positive, 100.0, 200.0, 200.0);
when xvFollower.Mechanism.Slope.IsAtConstantSpeed continue;
delay(t#300ms);

tecGear.MovePtp.StartSuperPose(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 40.0, 75.0,
200.0, 200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;
```



Legende ([siehe Legende Use Case 01](#))

Phasen:

- Die Leitachse fährt mit Zielgeschwindigkeit. Die Folgeachse ist eingekoppelt und fährt synchron zur Leitachse. Der Koppelstatus ist „*Eingekoppelt*“.
- An der Modulogrenze wird die überlagerte Endlospositionierung „*MoveVelocity.StartSuperPose()*“ gestartet.
Die Sollgeschwindigkeit der Folgeachse steigt von der Synchrongeschwindigkeit um die vorgegebene Zielgeschwindigkeit an.
Die vorher parallel zur Sollposition der Leitachse verlaufende Sollposition der Folgeachse wird steiler.
Der Rampenstatus wechselt von „angehalten“ über „beschleunigen“ auf „konstante Geschwindigkeit“.

- c) Nach 300 ms mit konstanter Geschwindigkeit wird die aktuell laufende überlagerte Endlospositionierung durch eine überlagerte Positionierung unterbrochen. Aufgrund einer niedrigeren Zielgeschwindigkeit verzögert zuerst die Folgeachse auf die neue Geschwindigkeit. Bei Erreichen der überlagerten Zielposition wechselt der Rampenstatus auf „*Angehalten*“. Die Geschwindigkeit der Folgeachse ist nun wieder gleich der Synchrongeschwindigkeit.
- d) Zum Ende des Anwendungsfalls wird die Leitachse gestoppt und die Folgeachse ausgekoppelt.

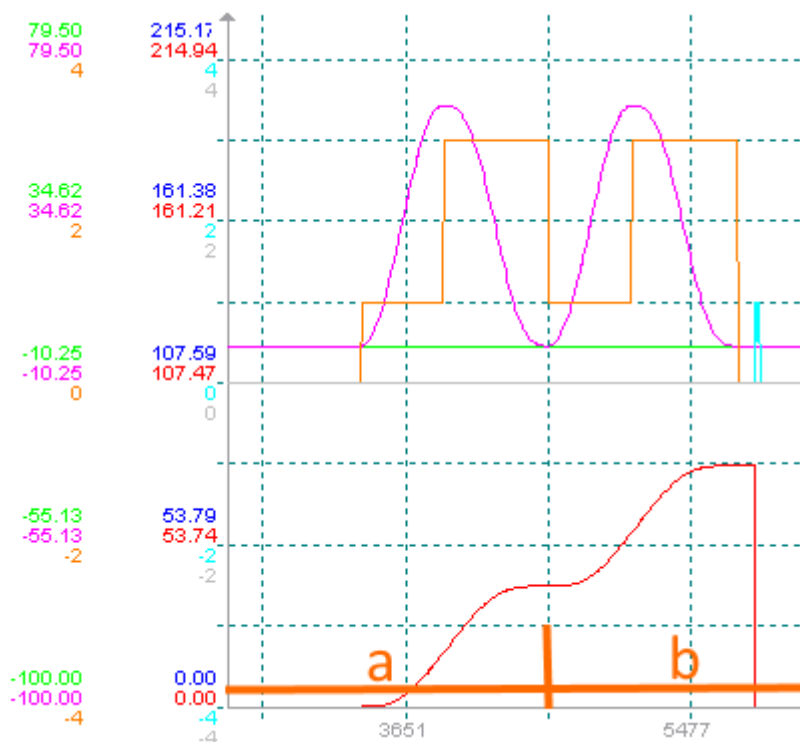
3.6.1.4 Use Case 04: Ausgekoppelt mit MovePtp.Start und MovePtp.StartSuperpose

In diesem Anwendungsfall soll gezeigt werden, dass sich im ausgekoppelten Zustand die Befehle „*MovePtp.Start*“ für eine normale Positionierung und „*MovePtp.StartSuperpose*“ für eine überlagerte Positionierung identisch verhalten.

Im ausgekoppelten Zustand der Folgeachse sind auch die Befehle „*MoveVelocity.Start*“ und „*MoveVelocity.StartSuperpose*“ identisch. Auf eine gesonderte Darstellung wird hier aber verzichtet.

```
tecGear.MovePtp.StartSuperPose(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 40.0, 100.0,
200.0, 200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;

tecGear.MovePtp.Start(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 80.0, 100.0, 200.0,
200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;
```



Legende ([siehe Legende Use Case 01](#))

Phasen:

- Der Zustand der Folgeachse ist ausgekoppelt.
Es wird eine überlagerte Positionierung auf eine absolute Zielposition gestartet.
Der Rampenstatus wechselt in diesem Fall von „*Beschleunigen*“ auf „*Verzögern*“.
Die Folgeachse hält bei der Position 40.0 an.
- Es wird eine normale Positionierung auf eine absolute Zielposition gestartet.
Der Rampenstatus wechselt in diesem Fall von „*Beschleunigen*“ auf „*Verzögern*“.
Die Folgeachse hält bei der Position 80.0 an.
Der Verlauf beider Positionierungen ist gleich.

3.6.1.5 Use Case 05: Ausgekoppelt mit MovePtp.StartSuperpose/MovePtp.Start gefolgt von Einkoppeln im Modus „Sofort“, Ausgekoppelt mit gefolgt von Einkoppeln im Modus „Sofort“

In diesem Anwendungsfall soll das Verhalten gezeigt werden, wenn während einer normalen oder überlagerten Positionierung mit dem Modus „Sofort“ eingekoppelt wird.

Aus in diesem Fall verhalten sich beide Positionierungen identisch.

Bei Einkoppeln im Modus „Sofort“ wechselt der Koppelstatus in den Zustand „Eingekoppelt“. Die laufende Positionierung wird in beiden Fällen als eine überlagerte Positionierung beendet.

Siehe auch Use Case 06, da sich dieses Verhalten zum Einkoppelmodus „Schnell“ unterscheidet.

```
when xvMaster.Position.Setpoint > 100.0 continue;

// Beginning the sequence with MovePtp.StartSuperPose
tecGear.MovePtp.StartSuperPose(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 40.0, 100.0,
200.0, 200.0);
when xvFollower.Mechanism.Slope.IsAtConstantSpeed continue;

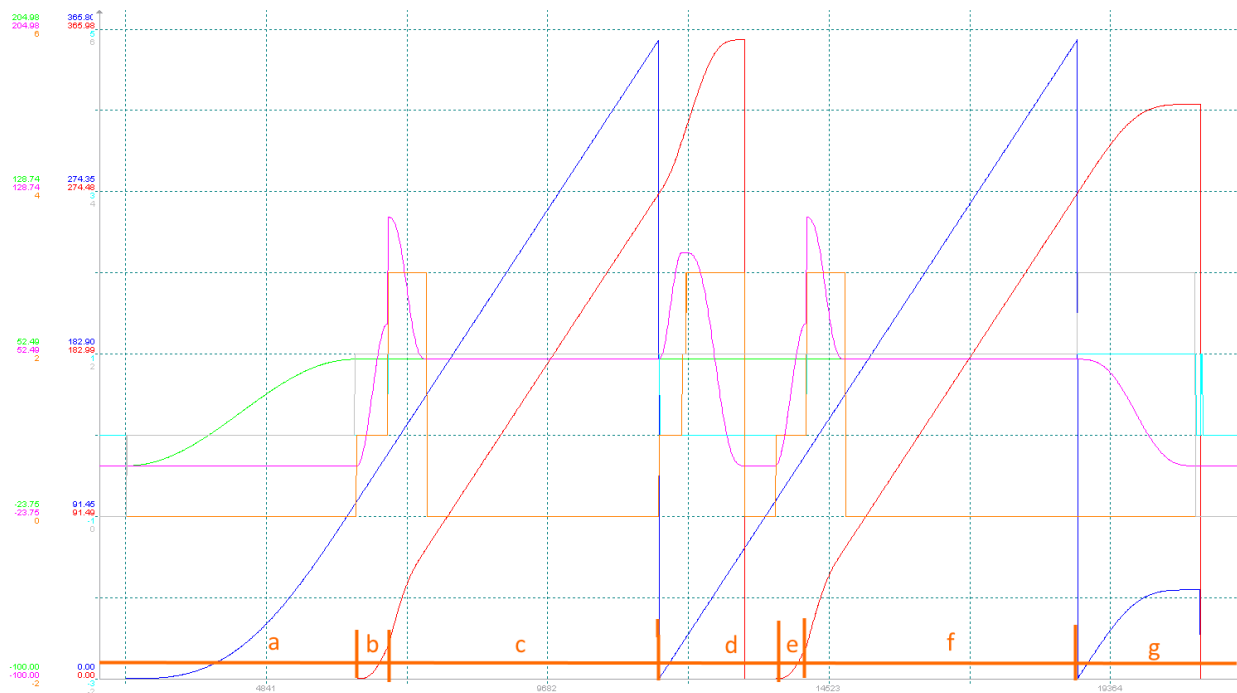
tecGear.Coupling.Couple(xvFollower, MCTechnoCoupleModes.Immediate);
when xvFollower.State.Techno.IsCoupled continue;

when xvMaster.Position.Setpoint < 100.0 continue;
// Decoupling and return to start position
tecGear.MovePtp.Start(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 0.0, 100.0, 200.0, 200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;

// Repeating sequence with MovePtp.Start
when xvMaster.Position.Setpoint < 100.0 continue;
when xvMaster.Position.Setpoint > 100.0 continue;

tecGear.MovePtp.Start(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 40.0, 100.0, 200.0,
200.0);
when xvFollower.Mechanism.Slope.IsAtConstantSpeed continue;

tecGear.Coupling.Couple(xvFollower, MCTechnoCoupleModes.Immediate);
when xvFollower.State.Techno.IsCoupled continue;
```



Legende ([siehe Legende Use Case 01](#))

Phasen:

- a) Die Leitachse beschleunigt auf Zielgeschwindigkeit. Die Folgeachse ist ausgekoppelt und steht.
- b) Es wird eine überlagerte Positionierung auf die Position 40.0 gestartet. Der Rampenstatus wechselt auf „Beschleunigen“.
- c) Bei Erreichen der Zielgeschwindigkeit der laufenden Positionierung wird „sofort“ eingekoppelt. Der Koppelstatus wechselt auf „eingekoppelt“. Da auf eine laufende Leitachse „sofort“ eingekoppelt wird, wird kein Einkoppelpolynom gerechnet, sondern die Folgeachse fährt auch „sofort“ mit der Synchrongeschwindigkeit. Dies ist ein Sprung im Geschwindigkeitssollwert. Dieser Sprung ist auch in der Sollgeschwindigkeit der Folgeachse zu sehen. Die nun überlagerte Positionierung wird beendet, wodurch die Geschwindigkeit bis auf die Synchrongeschwindigkeit sinkt.
- d) Die Folgeachse wird durch eine normale Positionierung ausgekoppelt und auf Startposition gefahren.
- e) Siehe auch b) nur mit den Unterschied, dass hier nun eine normale Positionierung gestartet wird.
- f) Siehe c).
- g) Zum Ende des Anwendungsfalls wird die Leitachse gestoppt und die Folgeachse ausgekoppelt.

3.6.1.6 Use Case 06: Ausgekoppelt mit MovePtp.StartSuperpose/MovePtp.Start gefolgt von Einkoppeln im Modus „Schnell“ oder „Wartend“

In diesem Anwendungsfall soll das Verhalten gezeigt werden, wenn während einer normalen oder überlagerten Positionierung mit dem Modus „Schnell“ eingekoppelt wird.

Das Verhalten im Modus „Wartend“ ist ähnlich dem Modus „Schnell“. Auf eine gesonderte Darstellung wird hier verzichtet.

Auch in diesem Fall verhalten sich beide Positionierungen identisch.

Im Gegensatz zum Use Case 05 wird nun die laufende Positionierung unterbrochen und mit einem Einkoppelpolynom entsprechend der Synchronposition auf die Leitachse aufsynchronisiert.

```
// Beginning the sequence with MovePtp.StartSuperPose
tecGear.MovePtp.StartSuperPose(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 40.0, 100.0,
200.0, 200.0);
when xvFollower.Mechanism.Slope.IsAtConstantSpeed continue;

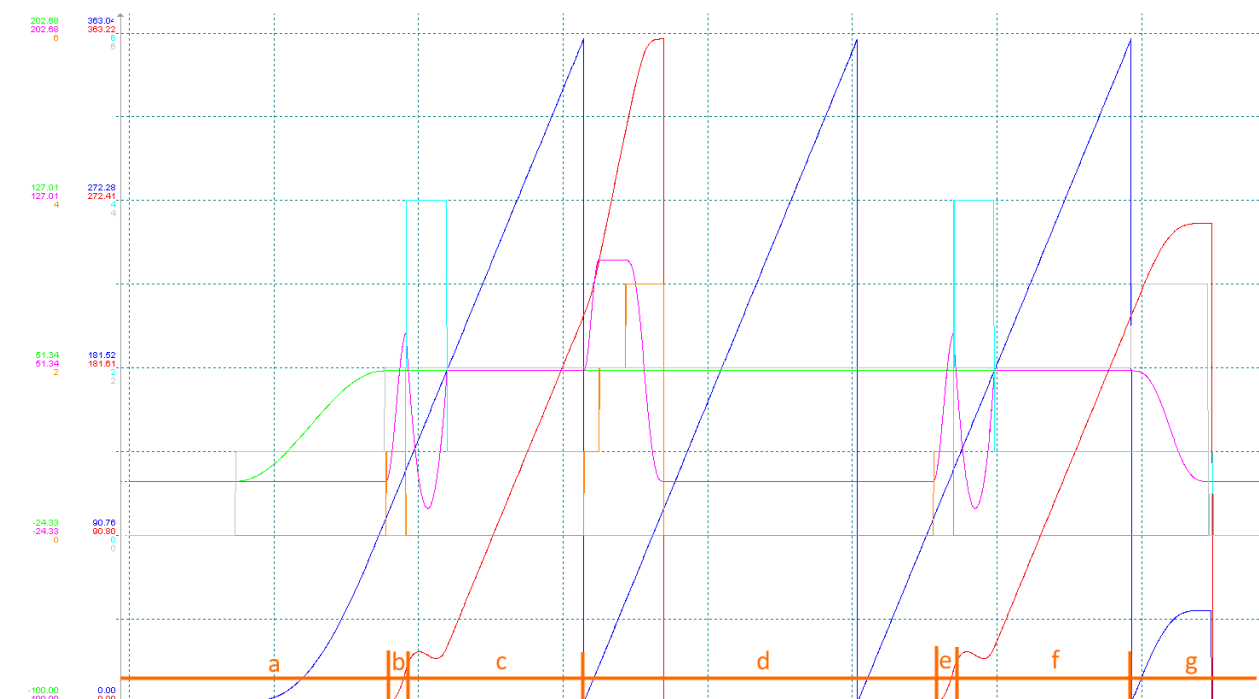
tecGear.Coupling.Couple(xvFollower, MCTechnoCoupleModes.Fast, 180.0, Directions.Positive, 30.0);
when xvFollower.State.Techno.IsCoupled continue;

when xvMaster.Position.Setpoint < 100.0 continue;
// Decoupling and returning to start position
tecGear.MovePtp.Start(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 0.0, 100.0, 200.0, 200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;

// Repeating sequence with MovePtp.Start
when xvMaster.Position.Setpoint < 100.0 continue;
when xvMaster.Position.Setpoint > 100.0 continue;

tecGear.MovePtp.Start(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 40.0, 100.0, 200.0,
200.0);
when xvFollower.Mechanism.Slope.IsAtConstantSpeed continue;

tecGear.Coupling.Couple(xvFollower, MCTechnoCoupleModes.Fast, 180.0, Directions.Positive, 30.0);
when xvFollower.State.Techno.IsCoupled continue;
```



Legende ([siehe Legende Use Case 01](#))

Phasen:

- a) Die Leitachse beschleunigt auf Zielgeschwindigkeit. Die Folgeachse ist ausgekoppelt und steht.
- b) Es wird eine überlagerte Positionierung auf die Position 40.0 gestartet.
Der Rampenstatus wechselt auf „Beschleunigen“.
- c) Bei Erreichen der Zielgeschwindigkeit der laufenden Positionierung wird „schnell“ eingekoppelt.
Der Koppelstatus wechselt auf „einkoppeln“.
Die Sollgeschwindigkeit springt nicht wie beim Einkoppelmodus „sofort“, sondern ein Einkoppelpolynom wird berechnet und die Folgeachse synchronisiert zum Synchronpunkt auf.
Die laufende überlagerte Positionierung wird direkt beendet.
- d) Die Folgeachse wird durch eine normale Positionierung ausgekoppelt und auf Startposition gefahren.
- e) Sieh auch b) nur mit den Unterschied, dass hier nun eine normale Positionierung gestartet wird.
- f) Siehe c).
- g) Zum Ende des Anwendungsfalls wird die Leitachse gestoppt und die Folgeachse ausgekoppelt.

3.6.1.7 Use Case 07: Einkoppeln im Modus „Schnell“ gefolgt von MovePtp.StartSuperpose/MovePtp.Start

In diesem Anwendungsfall wird nun in umgekehrter Reihenfolge als im Use Case 06 positioniert und eingekoppelt.

Hier soll ersichtlich werden, wie die Folgeachse reagiert, während sie sich in der Einkoppelbewegung befindet oder auf den optimalen Einkoppelpunkt (OCP) wartet und eine überlagerte bzw. normale Positionierung gestartet wird.

```
// Beginning sequence with MovePtp.StartSuperPose before OCP
tecGear.Coupling.Couple(xvFollower, MCTechnoCoupleModes.Fast, 180.0, Directions.Positive, 30.0);
when xvFollower.State.Techno.IsCoupling continue;
delay(t#10ms);
tecGear.MovePtp.StartSuperPose(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 40.0, 100.0,
200.0, 200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;
when xvMaster.Position.Setpoint > 250.0 continue;
// Decoupling and returning to start position
tecGear.MovePtp.Start(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 0.0, 100.0, 200.0, 200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;

// Repeating sequence with MovePtp.StartSuperPose between OCP and SP
when xvMaster.Position.Setpoint < 100.0 continue;
when xvMaster.Position.Setpoint > 100.0 continue;

tecGear.Coupling.Couple(xvFollower, MCTechnoCoupleModes.Fast, 180.0, Directions.Positive, 30.0);
when xvFollower.State.Techno.IsCoupling continue;
when xvMaster.Position.Setpoint > 130.0 continue;
tecGear.MovePtp.StartSuperPose(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 40.0, 100.0,
200.0, 200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;
when xvMaster.Position.Setpoint > 250.0 continue;
// Decoupling and returning to start position
tecGear.MovePtp.Start(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 0.0, 100.0, 200.0, 200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;

// Repeating sequence with MovePtp.Start before OCP
when xvMaster.Position.Setpoint < 100.0 continue;
when xvMaster.Position.Setpoint > 100.0 continue;

tecGear.Coupling.Couple(xvFollower, MCTechnoCoupleModes.Fast, 180.0, Directions.Positive, 30.0);
when xvFollower.State.Techno.IsCoupling continue;
delay(t#50ms);
tecGear.MovePtp.Start(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 40.0, 100.0, 200.0,
200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;
when xvMaster.Position.Setpoint > 250.0 continue;
// Decoupling and returning to start position
tecGear.MovePtp.Start(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 0.0, 100.0, 200.0, 200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;

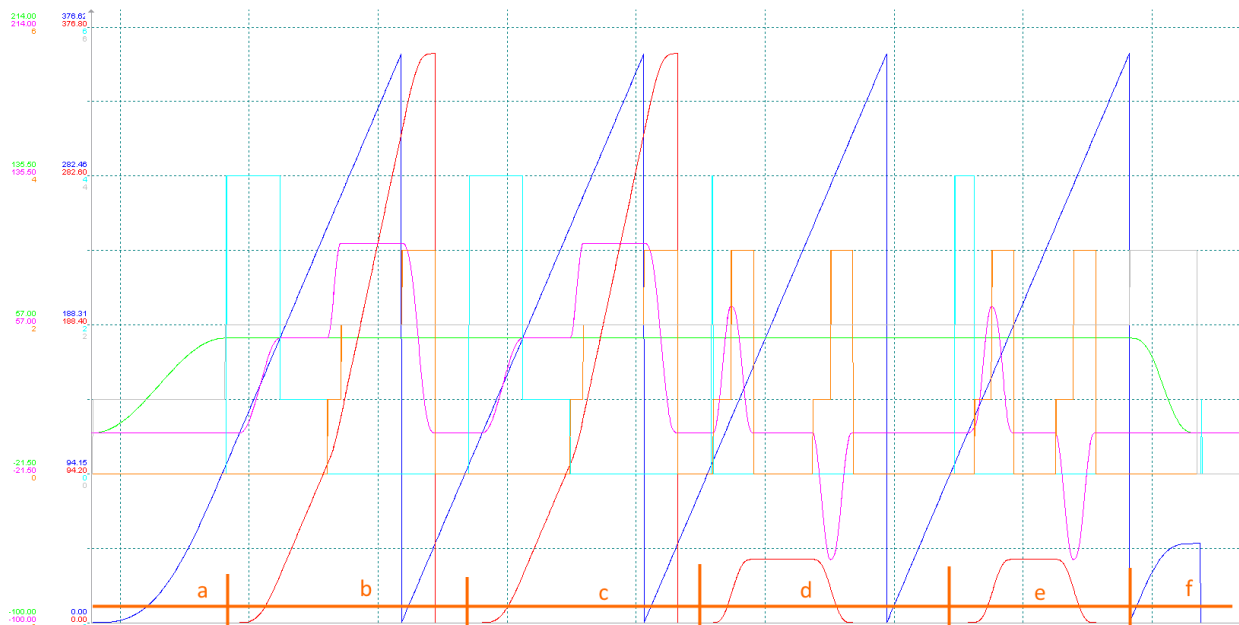
// Repeating sequence with MovePtp.Start between OCP and SP
when xvMaster.Position.Setpoint < 100.0 continue;
when xvMaster.Position.Setpoint > 100.0 continue;

tecGear.Coupling.Couple(xvFollower, MCTechnoCoupleModes.Fast, 180.0, Directions.Positive, 30.0);
when xvFollower.State.Techno.IsCoupling continue;
```

```

when xvMaster.Position.Setpoint > 130.0 continue;
tecGear.MovePtp.Start(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 40.0, 100.0, 200.0,
200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;
when xvMaster.Position.Setpoint > 250.0 continue;
// Decoupling and returning to start position
tecGear.MovePtp.Start(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 0.0, 100.0, 200.0, 200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;

```



Legende ([siehe Legende Use Case 01](#))

Phasen:

- Die Leitachse beschleunigt auf Zielgeschwindigkeit. Die Folgeachse ist ausgekoppelt und steht.
- Es wird mit Modus „Schnell“ eingekoppelt.
Noch bevor die Einkoppelbewegung beginnt wird eine überlagerte Positionierung gestartet.

```

16:37:23.197 tecGear / xvFollower: MotionMoveCouple ( ) - Aufruf ok
16:37:23.212 tecGear / xvFollower: Fehler: 7301 - Stufe: Meldung - Überlagerte Positionierung starten: Ein-
/Auskoppelvorgang läuft
16:37:23.212 tecGear / xvFollower: MotionMoveSuperPosePtp ( ) - Aufruf ok
16:37:26.197 tecGear / xvFollower: MotionMovePtp ( ) - Aufruf ok
16:37:30.398 tecGear / xvFollower: MotionMoveCouple ( ) - Aufruf ok
16:37:30.997 tecGear / xvFollower: Fehler: 7301 - Stufe: Meldung - Überlagerte Positionierung starten: Ein-
/Auskoppelvorgang läuft
16:37:30.998 tecGear / xvFollower: MotionMoveSuperPosePtp ( ) - Aufruf ok
16:37:33.398 tecGear / xvFollower: MotionMovePtp ( ) - Aufruf ok

```

Eine überlagerte Positionierung in dieser Phase wird abgewiesen und nicht ausgeführt!

- Es wird mit Modus „Schnell“ eingekoppelt.
Nachdem die Einkoppelbewegung beginnt, wird eine überlagerte Positionierung gestartet.
Auch in diesem Fall wird die überlagerte Positionierung abgewiesen und nicht ausgeführt!
- Es wird mit Modus „Schnell“ eingekoppelt.
Noch bevor die Einkoppelbewegung beginnt wird, eine normale Positionierung gestartet.

- Hierdurch wird der Einkoppelvorgang abgebrochen und eine Positionierung gestartet.
Der Koppelzustand wechselt von „Einkoppeln“ nach „Ausgekoppelt“.
- e) Es wird mit Modus „Schnell“ eingekoppelt.
Nachdem die Einkoppelbewegung beginnt wird eine normale Positionierung gestartet.
Hierdurch wird der Einkoppelvorgang abgebrochen und eine Positionierung gestartet.
Der Koppelzustand wechselt von „Einkoppeln“ nach „Ausgekoppelt“.
 - f) Zum Ende des Anwendungsfalls wird die Leitachse gestoppt.

3.6.1.8 Use Case 08: MovePtp.StartSuperpose und MoveVelocity.StartSuperpose während Auskoppeln

In diesem Anwendungsfall soll gezeigt werden, wie sich die Folgeachse verhält, wenn während des Auskoppelns eine überlagerte Positionierung bzw. Endlospositionierung gestartet wird. Es wird deutlich, dass überlagerte Positionierungen während des Auskoppelns abgewiesen und nicht ausgeführt werden.

Hierbei generiert die MCX die Fehlermeldung 7301, um dem Benutzer diesen Umstand mitzuteilen.

```
// Beginning sequence with MovePtp
tecGear.Coupling.Couple(xvFollower, MCTechnoCoupleModes.Fast, 180.0, Directions.Positive, 30.0);
when xvFollower.State.Techno.IsCoupled continue;

when xvMaster.Position.Setpoint > 250.0 continue;
tecGear.Coupling.Decouple(xvFollower, MCTechnoDecoupleModes.AtMasterPosition_Stop, 350.0, 1000.0);
when xvFollower.State.Techno.IsDecoupling continue;

tecGear.MovePtp.StartSuperPose(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 40.0, 100.0,
200.0, 200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;

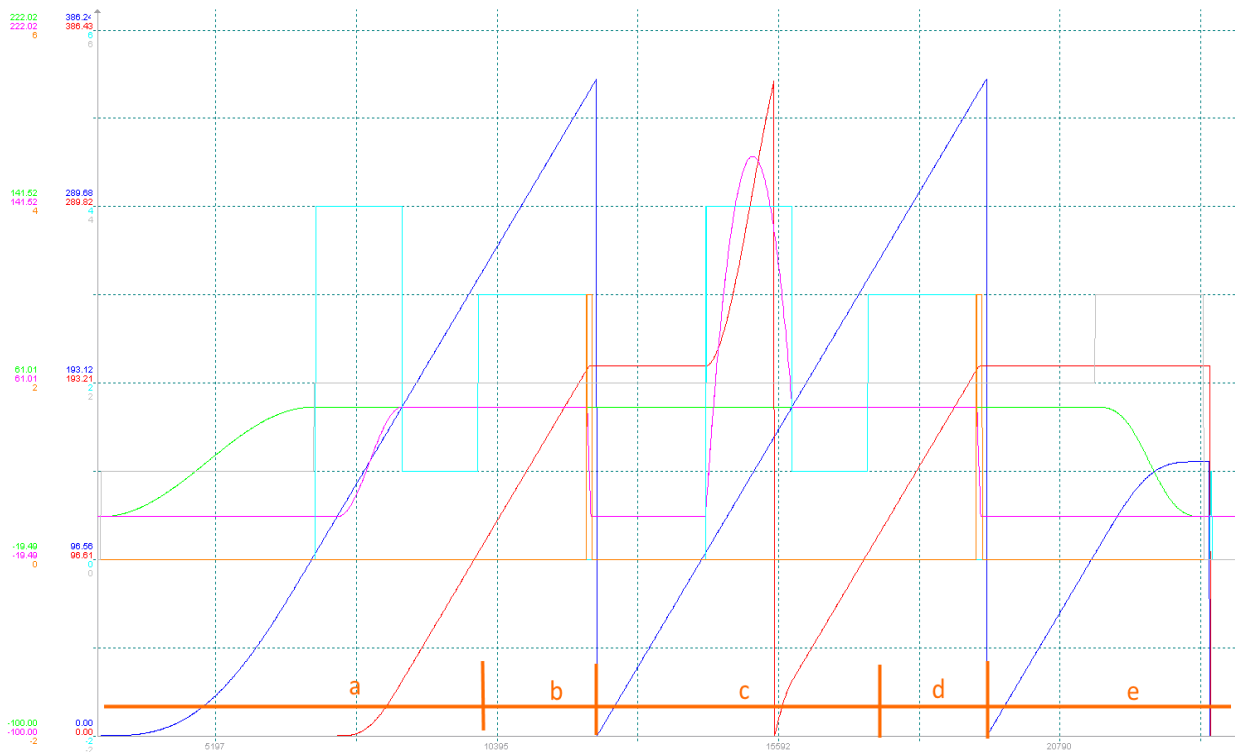
// Repeating sequence with MoveVelocity
when xvMaster.Position.Setpoint < 100.0 continue;
when xvMaster.Position.Setpoint > 100.0 continue;

tecGear.Coupling.Couple(xvFollower, MCTechnoCoupleModes.Fast, 180.0, Directions.Positive, 30.0);
when xvFollower.State.Techno.IsCoupled continue;

when xvMaster.Position.Setpoint > 250.0 continue;
tecGear.Coupling.Decouple(xvFollower, MCTechnoDecoupleModes.AtMasterPosition_Stop, 350.0, 1000.0);
when xvFollower.State.Techno.IsDecoupling continue;

tecGear.MoveVelocity.StartSuperPose(xvFollower, Directions.Positive, 100.0, 200.0, 200.0);

when xvMaster.Position.Setpoint < 100.0 continue;
when xvMaster.Position.Setpoint > 100.0 continue;
tecGear.MoveHalt.Start(xvFollower, MCTechnoHaltModes.Normal);
when xvFollower.Mechanism.Slope.IsStopped continue;
```



Legende ([siehe Legende Use Case 01](#))

Phasen:

- Die Leitachse beschleunigt auf Zielgeschwindigkeit. Die Folgeachse ist ausgekoppelt und steht.
Es wird mit Modus „Schnell“ eingekoppelt.
- Nach einer kurzen Synchronfahrt wird mit dem Modus „Stopp ab Leitachsenposition“ ausgekoppelt.
Der Koppelzustand wechselt auf „Auskoppeln“.
Es wird eine überlagerte Positionierung gestartet. Diese wird abgewiesen und nicht ausgeführt.

```

16:42:35.362 tecGear / xvMaster: MotionMoveVelocity ( ) - Aufruf ok
16:42:39.325 tecGear / xvFollower: MotionMoveCouple ( ) - Aufruf ok
16:42:42.326 tecGear / xvFollower: MotionMoveDecouple ( ) - Aufruf ok
16:42:42.330 tecGear / xvFollower: Fehler: 7301 - Stufe: Meldung - Überlagerte Positionierung starten: Ein-/Auskoppelvorgang läuft
16:42:42.330 tecGear / xvFollower: MotionMoveSuperPosePtp ( ) - Aufruf ok
16:42:46.526 tecGear / xvFollower: MotionMoveCouple ( ) - Aufruf ok
16:42:49.525 tecGear / xvFollower: MotionMoveDecouple ( ) - Aufruf ok
16:42:49.530 tecGear / xvFollower: Fehler: 7301 - Stufe: Meldung - Überlagerte Positionierung starten: Ein-/Auskoppelvorgang läuft
16:42:49.530 tecGear / xvFollower: MotionMoveSuperPoseVelocity ( ) - Aufruf ok
16:42:53.725 tecGear / xvFollower: MotionStop ( ) - Aufruf ok
  
```

Deshalb wird wie im Auskoppelkommando angegeben an der Leitachsposition ausgekoppelt und mit der eingestellten Verzögerung angehalten.

- Die Folgeachse wird wieder im Modus „Schnell“ eingekoppelt.
- Nach einer kurzen Synchronfahrt wird mit dem Modus „Stopp ab Leitachsenposition“ ausgekoppelt.
Der Koppelzustand wechselt auf „Auskoppeln“.
Es wird eine überlagerte Endlospositionierung gestartet. Diese wird abgewiesen und nicht ausgeführt.

Deshalb wird wie im Auskoppelkommando angegeben an der Leitachsisposition ausgekoppelt und mit der eingestellten Verzögerung angehalten.

- e) Zum Ende des Anwendungsfalls wird die Leitachse gestoppt.

3.6.1.9 Use Case 09: Auskoppeln während MovePtp.StartSuperpose und MoveVelocity.StartSuperpose

In diesem Anwendungsfall soll im Gegensatz zum Use Case 08 gezeigt werden, wie sich die Folgeachse verhält, wenn während einer überlagerten Positionierung ausgekoppelt wird. Es soll gezeigt werden, dass in diesem Fall die überlagerte Positionierung abgebrochen wird und direkt ausgekoppelt wird.

```
// Beginning sequence with MovePtp
tecGear.Coupling.Couple(xvFollower, MCTechnoCoupleModes.Fast, 180.0, Directions.Positive, 30.0);
when xvFollower.State.Techno.IsCoupled continue;

tecGear.MovePtp.StartSuperPose(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 40.0, 100.0,
200.0, 200.0);
when xvFollower.Mechanism.Slope.IsAtConstantSpeed continue;

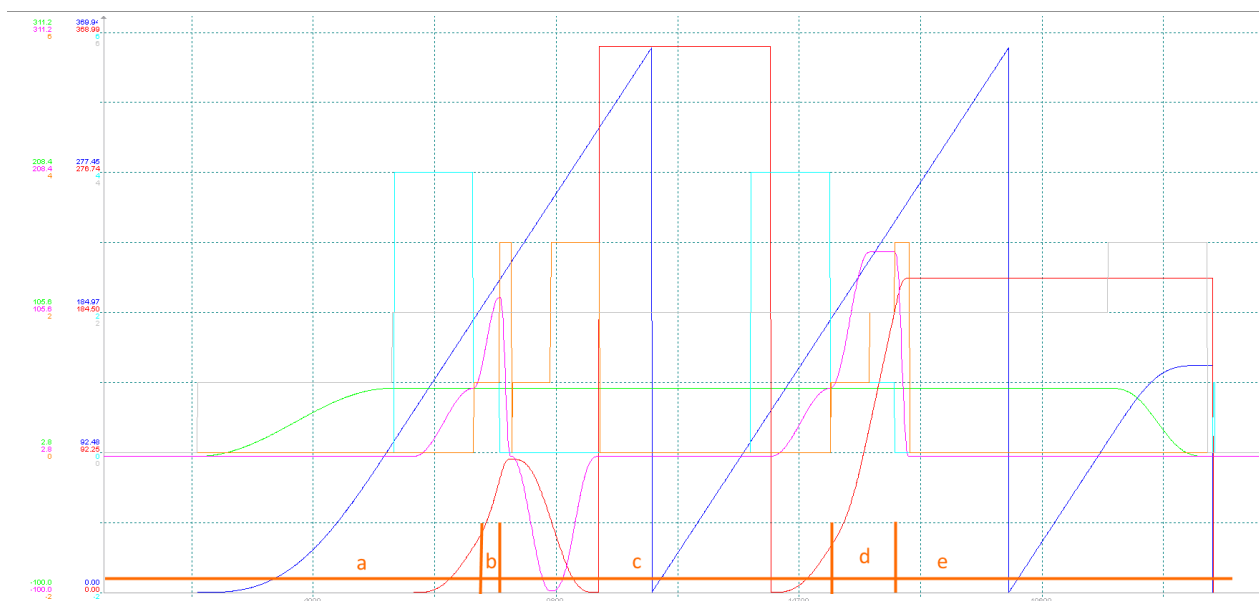
tecGear.Coupling.Decouple(xvFollower, MCTechnoDecoupleModes.Immediate_Stop, , 1000.0);
when xvFollower.State.Techno.IsDecoupled and xvFollower.Mechanism.Slope.IsStopped continue;

tecGear.MovePtp.Start(xvFollower, MCTechnoPositioningModes.AbsModuloAuto, 0.0, 100.0, 200.0, 200.0);
when xvFollower.Mechanism.Slope.IsStopped continue;

// Repeating sequence with MoveVelocity
when xvMaster.Position.Setpoint < 100.0 continue;
when xvMaster.Position.Setpoint > 100.0 continue;

tecGear.Coupling.Couple(xvFollower, MCTechnoCoupleModes.Fast, 180.0, Directions.Positive, 30.0);
when xvFollower.State.Techno.IsCoupled continue;
    tecGear.MoveVelocity.StartSuperPose(xvFollower, Directions.Positive, 100.0, 200.0, 200.0);
when xvFollower.Mechanism.Slope.IsAtConstantSpeed continue;
delay(t#500ms);
tecGear.Coupling.Decouple(xvFollower, MCTechnoDecoupleModes.Immediate_Stop, , 1000.0);
when xvFollower.State.Techno.IsDecoupled continue;

when xvMaster.Position.Setpoint < 100.0 continue;
when xvMaster.Position.Setpoint > 100.0 continue;
tecGear.MoveHalt.Start(xvFollower, MCTechnoHaltModes.Normal);
when xvFollower.Mechanism.Slope.IsStopped continue;
```



Legende ([siehe Legende Use Case 01](#))

Phasen:

- a) Die Leitachse beschleunigt auf Zielgeschwindigkeit. Die Folgeachse ist ausgekoppelt und steht.
Im weiteren Verlauf wird die Folgeachse eingekoppelt.
Der Koppelstatus wechselt auf „einkoppeln“. Die Sollgeschwindigkeit der Folgeachse steigt bis auf Synchrongeschwindigkeit an.
- b) Sobald die Folgeachse eingekoppelt ist, startet eine überlagerte Positionierung.
Der Rampenstatus wechselt auf „beschleunigen“.
- c) Sobald der Rampenstatus auf „konstante Geschwindigkeit“ wechselt, wird die Folgeachse im Modus „Sofort mit Anhalten“ ausgekoppelt.
Der Koppelstatus wechselt auf „ausgekoppelt“.
Der Rampenstatus wechselt auf „verzögern“, da die Folgeachse direkt mit der eingestellten Verzögerung gestoppt wird.
Nach dem Anhalten wird die Folgeachse wieder auf Grundposition gefahren.
Im weiteren Verlauf wird die Folgeachse wieder eingekoppelt.
Der Koppelstatus wechselt auf „einkoppeln“. Die Sollgeschwindigkeit der Folgeachse steigt bis auf Synchrongeschwindigkeit an.
- d) Sobald die Folgeachse eingekoppelt ist, startet eine überlagerte Endlospositionierung.
Der Rampenstatus wechselt auf „beschleunigen“, und weiter auf „konstante Geschwindigkeit“.
- e) Nach einer kurzen Wartezeit wird die Folgeachse im Modus „Sofort mit Anhalten“ ausgekoppelt.
Der Koppelstatus wechselt auf „ausgekoppelt“.
Der Rampenstatus wechselt auf „verzögern“, da die Folgeachse direkt mit der eingestellten Verzögerung gestoppt wird.

Jetter AG
Gräterstraße 2
71642 Ludwigsburg
www.jetter.de

E-Mail info@jetter.de
Telefon +49 7141 2550-0

We automate your success.