

Topic:

- How does IPM (= Interpolated Position Mode) work in detail?
- What is important to take care about when IPM is in use?
- How can PVT data points be calculated in case of an existing function (e.g. sinus)?

Technical background:

How is the interpolation processed by IPM?

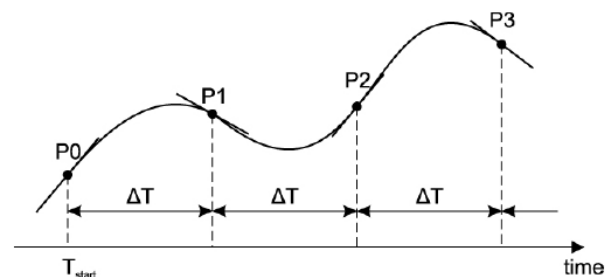
Why there is a set of 3 data values (PVT) required for each curve point (and not just 2)?

IPM (= Interpolated Position Mode) calculates the **motion profile (= trajectory generation) in between transferred position points**. The generated interpolated motion path feeds the internal position control loop with new position set values every 1 ms (= 1 kHz rate) taking the velocity, distance, and time of the next configured curve data into account. The interpolation requires the following three values: Position (P), Velocity (V), and time (T resp. delta-T). These are the so-called **PVT values (= Position, Velocity, Time)**.

Position [qc]: Next absolute PVT position, which have to be reached
P0, P1, P2, P3, ...

Velocity [rpm]: Velocity of the next PVT data set, i.e. derivative of the position
-> Tangent of the curve position
-> Gradient

Time [ms]: Time span (-> delta-T) in between the last and next PVT point. It is possible to configure different periods of time for each PVT data set.



The motion calculation in between consecutive PVT data sets is based on a **3rd order interpolation**. This means that there is no linear connection in between the position points of a PVT data set active but a cubic spline curve applied generating the required demand position data for each position control cycle. This means that there are **3 values required for the interpolation**. These are the so-called **PVT (= Position, Velocity, Time) values**.

What can be seen by the graph above:

- P0 ... P3 are configured curve position values which have to be passed by.
- delta-T is the time span of the motion in between consecutive curve position.
- The tangents of each curve points correspond to the velocity passing by the curve point.

maxon motor control

maxon motor ag Brünigstrasse 220 CH – 6072 Sachseln www.maxonmotor.com	EPOS2: Interpolated Position Mode (IPM) Practical usage and configuration	Version: 1.10 (En.) Author: MANA / WJ Date: 2018-09-06
---	--	--

What are possible values for delta-T?

It is possible to choose delta-T values in the range of 1 ms bis 255 ms in case of an EPOS2 controller. From the theoretical point of view the motion path is defined in a more precise way and the position accuracy is improved in case of small delta-T values. This also means that a master has to transmit more PVT data values per time which results in a higher communication load. The EPOS2 has to interpolate consecutive PVT values based on just small time spans. Finally there are some limitations from the practical point of view which have to be taken into account too:

The communication rate for transmitting PVT data depends on the communication interface in use. It is important that delta-T is not smaller than the cycle time which is required by the interface to transmit next PVT data fast enough in a reliable way.

Typical cycle rate for the transmission of one PVT data set:

Communication interface	Minimum cycle time required to transmit one PVT data set	Minimum possible delta-T
CAN (1 Mbit/s)	1 ms	delta-T > 1 ms
USB (EposCmd.DLL Library)	20 ms	delta-T > 20 ms
RS232 (EposCmd.dll, 115'200 Bit/s)	30 ms	delta-T > 30 ms

Remarks:

- The USB and RS232 cycle times mentioned in the table are typical values experienced by a PC based master and application software using the EPOS command library (EposCmd.dll). It is possible that even **longer delta-T values are required depending on the load and performance of the master** (= PC) in use. There will be an „Underflow“ error of the IPM data buffer occur if the master is not able to transmit the data fast enough to ensure that there is always the next PVT data set present in time for interpolation. Additional information about the expected performance and cycle times of USB communication based on the EPOS Command Library (= EposCmd.dll) is available by the following Support Center document too: [„EPOS Command Library: Data exchange rates“](#)
- A delta-T value of 1 ms is possible in case of CAN communication (1 Mbit/s, PDO) by a real time master (e.g. PLC) but not necessary for IPM. Extreme low delta-T values in the range of 1 – 3 ms are no advantage in practice concerning improved position accuracy. Due to this there is the recommendation to use **delta-T values of 5 ms or bigger even if CAN communication is present**. You have to be aware that the EPOS2 position control loop is just processed every 1 ms. This also means that there is no time left processing interpolated position data in between consecutive PVT data sets in case of a 1 ms delta-T. If there is real time master able to calculate and transfer position data every 1 ms (using synchronized CANopen PDO data exchange), it is recommended to use the so-called “Position Mode” of the EPOS2 instead of IPM.
- **Important:**
It is required to **set delta-T = 0** of the last PVT data set to **stop IPM motion**.

Solution:

Calculation of PVT data sets in case of a sine wave movement:

It is not such a simple task to calculate the PVT data sets of a profiled motion. If the motion is specified by a mathematical function (e.g. position $p = \sin(t)$), everything gets much easier because the velocity and acceleration at each profile point can be calculated by the derivation $v = dp/dt$ and $a = dp/dv = dp^2/dt^2$:

- *Position* [qc]: $p = \sin(t)$
- *Velocity* $\left[\frac{qc}{s}\right]$: $v = \frac{d}{dt}\sin(t) = \cos(t)$
- *Acceleration* $\left[\frac{qc}{s^2}\right]$: $a = \frac{d^2}{dt^2}\sin(t) = -\sin(t)$

The motor will be in standstill when the IPM motion starts. Therefore it is important that the velocity value of the initial PVT data set is also 0 or almost 0. An initial PVT velocity value bigger than 0 results in an extreme high acceleration at the start of the motion which triggers the „Max. Acceleration“ IPM error bit of the status word. This can be prevented if the velocity function $\cos(t)$ is shifted by 90° to the right so that the motion starts with a velocity of 0 rpm.

This results in the following: $\cos\left(t - \frac{\pi}{2}\right) \Rightarrow \cos\left(t_0 - \frac{\pi}{2}\right) \Rightarrow \cos\left(0 - \frac{\pi}{2}\right) = 0$

The amplitude A [qc] and the frequency f [Hz] also has to be taken into account next. This results in the following function then:

- *Position* [qc]: $p = A * \sin\left(t * 2 * \pi * f - \frac{\pi}{2}\right)$
- *Velocity* $\left[\frac{qc}{s}\right]$: $v = \frac{d}{dt} A * \sin\left(t * 2 * \pi * f - \frac{\pi}{2}\right) = A * 2 * \pi * f * \cos\left(t * 2 * \pi * f - \frac{\pi}{2}\right)$
- *Acceleration* $\left[\frac{qc}{s^2}\right]$: $a = \frac{d^2}{dt^2} A * \sin\left(t * 2 * \pi * f - \frac{\pi}{2}\right) = A * 4 * \pi^2 * f^2 * -\sin\left(t * 2 * \pi * f - \frac{\pi}{2}\right)$

The EPOS2's velocity scaling is [rpm], i.e. the velocity value has to be converted:

$$[rpm] = 60 / (4 * \text{encoder resolution}) * [qc/s]$$

It is necessary to shift the sine curve one amplitude higher (+A) to get a starting position of 0 qc.

1. $p = A * \sin\left(t * 2 * \pi * f - \frac{\pi}{2}\right) + A [qc]$
2. $v = A * 2 * \pi * f * \cos\left(t * 2 * \pi * f - \frac{\pi}{2}\right) * \frac{60}{4 * \text{EncoderResolution}} [rpm]$
3. $a = A * 4 * \pi^2 * f^2 * -\sin\left(t * 2 * \pi * f - \frac{\pi}{2}\right) * \frac{60}{4 * \text{EncoderResolution}} \left[\frac{rpm}{s}\right]$

Finally t will be the time summed up of all delta-T.

maxon motor control

maxon motor ag
Brünigstrasse 220
CH – 6072 Sachseln
www.maxonmotor.com

EPOS2: Interpolated Position Mode (IPM) Practical usage and configuration

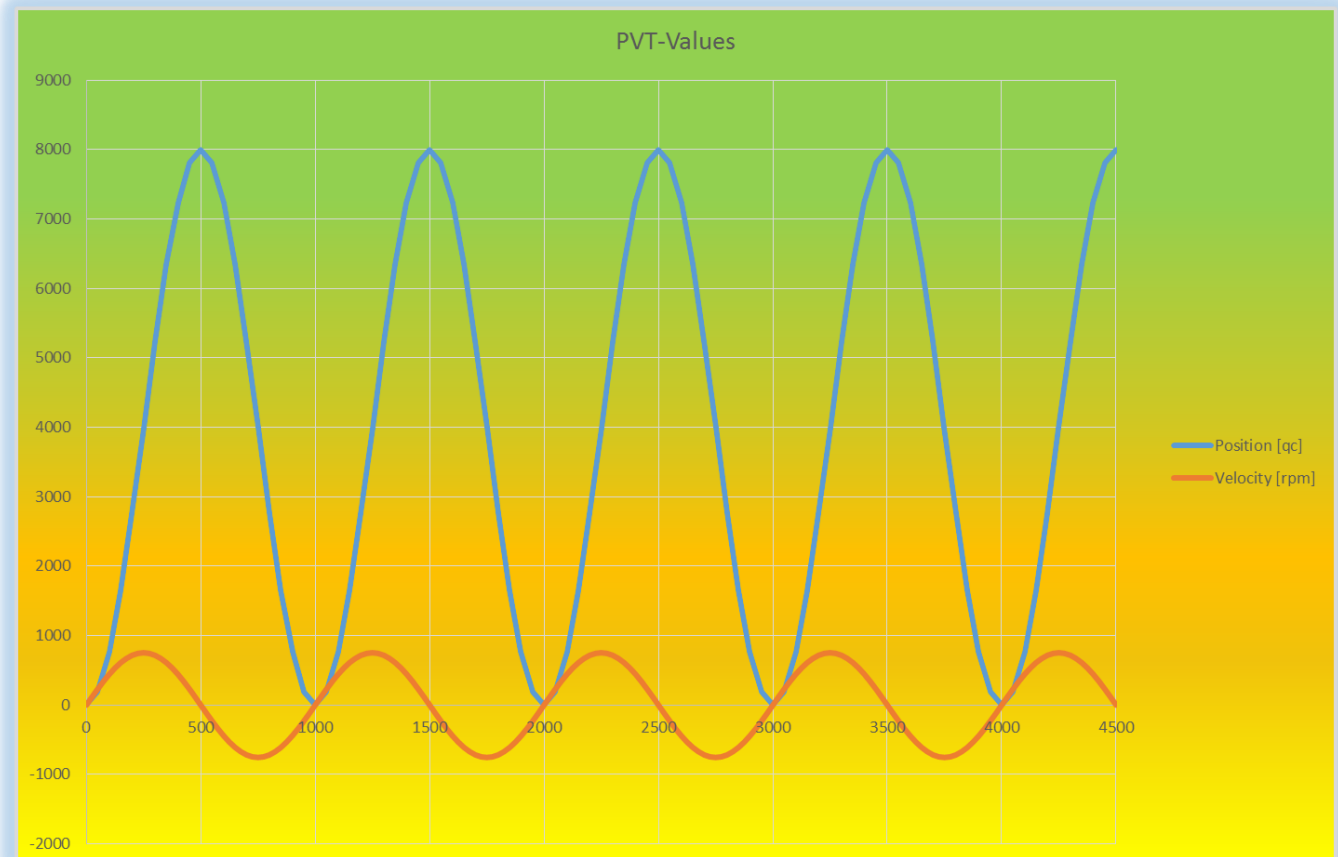
Version: 1.10 (En.)
Author: MANA / WJ
Date: 2018-09-06

Example:

delta-T = 50 ms, Amplitude A = 4000 qc, Frequency f = 1 Hz, Encoder resolution = 500 cpt

Zeit [s]	Position [qc]	Geschwindigkeit [rpm]	Beschleunigung [rpm/s]	Delta T [ms]
$t_0 = 0$	0	0	4737	50
$t_1 = 0.050$	196	233	4506	50
$t_2 = 0.100$	764	443	3833	50
$t_3 = 0.150$	1649	610	2785	50
...
$t_x = \dots$	0	0	0	0

The values above result in the following position and velocity curves then:



**„Interpolation Position Mode“ Error:
Why there is a risk to hit the system limits?**

If you take the example of a sinusoidal motion profile again, there is the possibility that too high amplitude or frequency values have been chosen. The result might be that required velocity or acceleration exceeds the configured or possible maximum velocity or acceleration of a system and drive unit. The „Interpolation Position Mode“ error reports that the configured limits are hit.

Based on the example’s data the velocity and acceleration will reach the following maximum:

- $V_{max} = A * 2 * \pi * f * \frac{60}{4 * EncoderResolution} = 4000 * 2 * \pi * 1 * \frac{60}{4 * 500} = 754 \text{ rpm}$
- $a_{max} = A * 4 * \pi^2 * f^2 * \frac{60}{4 * EncoderResolution} = 4000 * 4 * \pi^2 * 1^2 * \frac{60}{4 * 500} = 4'737 \frac{\text{rpm}}{\text{s}}$

If the frequency is increased from 1 Hz up to 5 Hz and the amplitude is increased from 4000 qc up to 8000 qc too, the resulting maximum velocity and acceleration gets much higher:

- $V_{max} = A * 2 * \pi * f * \frac{60}{4 * EncoderResolution} = 8000 * 2 * \pi * 5 * \frac{60}{4 * 500} = 7'540 \text{ rpm}$
- $a_{max} = A * 4 * \pi^2 * f^2 * \frac{60}{4 * EncoderResolution} = 8000 * 4 * \pi^2 * 5^2 * \frac{60}{4 * 500} = 236'857 \frac{\text{rpm}}{\text{s}}$

Even small changes can result in hitting or exceeding the configured system limits of „Maximal Profile Velocity“ (Object: 0x607F) or „Max Acceleration“ (Object: 0x60C5). It is recommended always to calculate the resulting maximum velocity and acceleration roughly and these values have to be compared with the configured limits and the technical data of the drive unit and mechanics.

**Hint:
Visual check by “EPOS Studio”**

It is possible to insert calculated PVT values or even import these as a *.CSV file into the “Interpolated Position Mode” tool of “EPOS Studio”. The tabs “Position Velocity” and “Acceleration” show the curves. A simple visual check is possible then: There should be no abrupt or steep rise of the velocity or acceleration present.

Analyzing the error state: “Interpolated Position Mode” Error (ErrorCode 0xFF0C):

One common error message during IPM operation is the “Interpolated Position Mode” error with the code 0xFF0C. It is necessary to check the content of the object “Interpolation Buffer Status” (0x20C4/01) to evaluate the root cause of this error state, e.g. by using the “Object Dictionary” tool of „EPOS Studio“. The bits 8 – 11 of the buffer status provide some information about the root cause of the IPM error 0xFF0C. There is a summary of the meaning of the bits present by chapter “8.2.70 Interpolation Buffer” of the document “EPOS2 Firmware Specification”.

Example: Object 0x20C4/01 = 0x410D

Binary representation of the data value 0x410D: 0100 0001 0000 1101

=> Bit 8 is set

=> Underflow Error

i.e. there was no next PVT data set value in time to calculate the interpolation for the next part of the motion.

maxon motor control

maxon motor ag
Brünigstrasse 220
CH – 6072 Sachseln
www.maxonmotor.com

EPOS2: Interpolated Position Mode (IPM) Practical usage and configuration

Version: 1.10 (En.)
Author: MANA / WJ
Date: 2018-09-06

Extract of chapter “8.2.70 Interpolation Buffer” of the “EPOS2 Firmware Specification”:

Name	Interpolation Buffer
Index	0x20C4
Number of entries	3

Name	Interpolation Buffer Status
Index	0x20C4
Subindex	0x01
Type	UNSIGNED16
Access	RO
Default Value	–
Value Range	–

Bit 15	Bit 14	Bit 13...12	Bit 11...8	Bit 7...4	Bit 3...0
IP Mode active	Buffer enabled	reserved (0)	IPM buffer errors	reserved (0)	IPM buffer warnings

Name	Bit	Value	Description
Underflow Warning	0	0	No buffer underflow warning
		1	Buffer underflow warning level (0x20C4-2) is reached
Overflow Warning	1	0	No buffer overflow warning
		1	Buffer overflow warning level (0x20C4-3) is reached
Velocity Warning	2	0	No profile velocity violation detected
		1	IPM velocity greater than profile velocity (0x6081) detected
Acceleration Warning	3	0	No profile acceleration violation detected
		1	IPM acceleration greater than profile acceleration (0x6083) detected
Underflow Error	8	0	No buffer underflow error
		1	Buffer underflow error (trajectory abort)
Overflow Error	9	0	No buffer overflow error
		1	Buffer overflow error (trajectory abort)
Velocity Error	10	0	No maximal profile velocity error
		1	IPM velocity greater than maximal profile velocity (0x607F) detected
Acceleration Error	11	0	No maximal profile acceleration error
		1	IPM acceleration greater than maximal profile acceleration (0x60C5) detected

IPM-Fehler und ihre Ursachen:

- Underflow Error:** The “Underflow Error” states that there are no more PVT data sets present in the EPOS2’s buffer to process the IPM calculation of the next motion step. This means that the PVT data sets are processed faster than these have been transmitted into the buffer by the master. The number of requested and processed PVT data sets per time just depends on delta-T. The master must be able to transmit PVT data sets fast enough into the buffer.
- Increase delta-T,
i.e. recalculate the PVT data sets on the base of a longer delta-T.
This reduces the number of required PVT data sets per time.
 - Use a bus interface (e.g. CAN instead of USB) with a better performance which is able to transmit the required PVT data sets fast enough.
 - Increase the CAN bitrate if possible,
i.e. the period of time to transmit the PVT data sets is slightly reduced then.
- Overflow Error:** The “Overflow Error” states that more PVT data sets have been transmitted to the EPOS2 than the internal circular buffer can take. This means that the master transmits more PVT data sets than actually required by the motion calculation and able to be captured by the circular buffer.
- Reduce delta-T,
i.e. recalculate the PVT data sets on the base of a shorter delta-T value.
This increases the number of required PVT data sets per time and also results in a more accurate definition of the motion profile.
 - Reduce the transmission rate of new PVT data sets,
i.e. increase the period in between two PVT data sets transmitted by the master.
- Velocity Error:** The “Velocity Error” states that the configured “Maximal Profile Velocity” (0x607F) has to be exceeded to follow the resulting motion profile based on the PVT data sets.
- Recalculate the PVT data sets to ensure that the configured “Maximal Profile Velocity” is not exceeded anymore by the resulting IPM motion profile.
 - Increase the configured “Maximal Profile Velocity” (0x607F) if the electrical and mechanical drive system can cope with higher speeds without the risk of damage.
- Acceleration Error:** The Acceleration Error states that the configured “Max Acceleration” (0x60C5) has to be exceeded to follow the resulting motion profile based on the PVT data sets.
- Recalculate the PVT data sets to ensure that the configured “Max Acceleration” is not exceeded anymore by the resulting IPM motion profile.
 - Increase the configured “Max Acceleration” (0x60C5) if the electrical and mechanical drive system can cope with higher speeds without the risk of damage.

Conclusion:

- Usage of IPM (= Interpolation Position Mode) is just recommended if a **required motion profile is known in advance** so that PVT data sets describing the motion profile can be calculated and buffered.
- It has to be checked that the resulting motion profile based on the PVT data sets **does not exceed the configured maximum acceleration and velocity limits** of the drive system:
 - ⇒ “Maximal Profile Velocity”, 0x607F
 - ⇒ “Max Acceleration”, 0x60C5
- The possible **value range of delta-T** (= period of time in between 2 PVT data sets) has to be noticed:
 - ⇒ $1\text{ms} < \text{delta-T} < 255\text{ms}$
- The communication rate of the bus interface and the actually possible **transmission rate of PVT data sets** (based on the master’s and bus performance) has to be taken into account by the definition of delta-T:
 - ⇒ Rule of thumb:
USB: $\text{delta-T} > 20\text{ms}$ / RS232: $\text{delta-T} > 30\text{ms}$ / CAN: $\text{delta-T} > 5\text{ms}$
- The EPOS2’s position control rate is 1 kHz, i.e. the delta-T value should **not be less than 5 ms in practice**. This ensures that there is actually an interpolation possible in between PVT data sets.
- “Buffer Underflow” and “Buffer Overflow” has to be avoided:
 - ⇒ delta-T has to be chosen in a way so that the **master is able to transmit new PVT data sets into the buffer at a sufficient rate** but the buffer should **not be flooded by too many data sets**.
- The maximum IPM buffer size of the EPOS2 has to be taken into account. Otherwise there is the risk that a “Buffer Overflow” error takes part.
 - ⇒ **64 PVT data sets** in maximum are possible by the IPM buffer (FIFO)
- The velocity V is calculated by the derivation of the position.
- The position value of the initial PVT data set has to correspond to the current motor position at IPM start.
- The processing time of the PVT data sets and the motion profile depends on the delta-T value.
- The encoder resolution influences the resulting maximum velocity and maximum acceleration. A lower encoder resolution results in a higher risk to exceed configured velocity and acceleration limits.
- It is strongly recommended to **check the IPM graphics**. There should be **no steep or abrupt changes of the velocity or acceleration visible**:
 - ⇒ EPOS Studio -> Tool -> „Interpolated Position Mode“ ->Tab: „Position Velocity / Acceleration“
- The PVT data sets just hold **absolute position values**.
- The **delta-T value** of the last PVT data set has to be **set to ‘0’ to stop the IPM motion profile**.
- **More information required** about EPOS2’s Interpolated Position Mode:
 - ⇒ **“EPOS2 Firmware Specification”**: Chapter 5.5 „Interpolation Position Mode“
 - ⇒ **“EPOS2 Application Notes Collection”**: Chapter 6 „Interpolation Position Mode“