

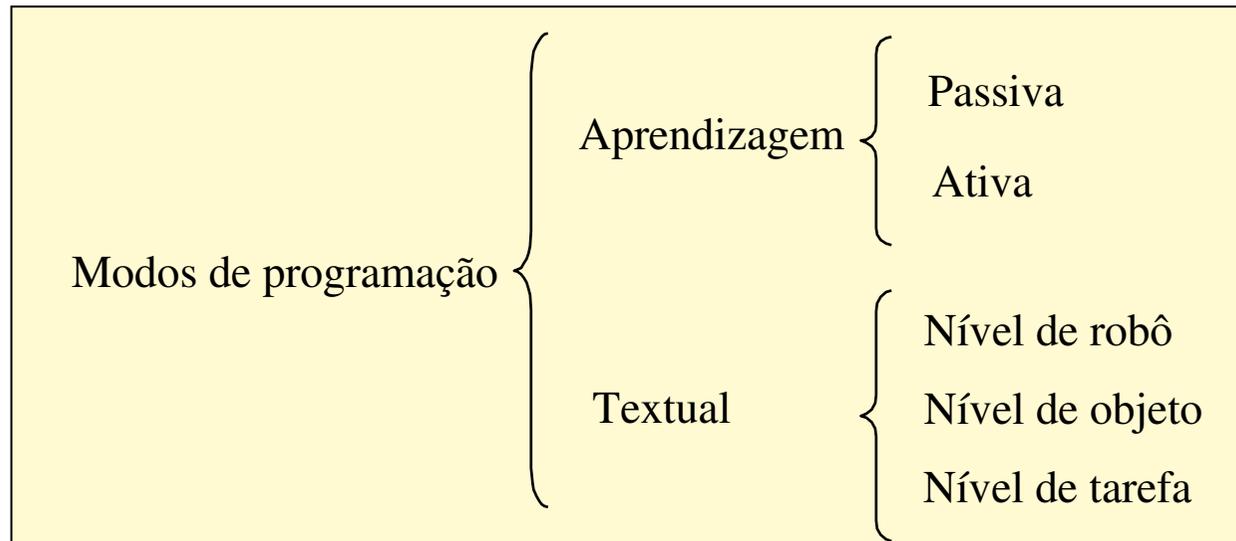
Programação de Robôs Industriais

Definição:

- A programação de um robô pode ser definida como o processo mediante o qual se indica a seqüência de ações que o robô deverá cumprir durante a realização de uma dada tarefa.

- Existe uma normalização para os procedimentos de programação, **Norma ISO TR 10562**.
- Entretanto, cada fabricante desenvolveu seu próprio método, válido unicamente para seus próprios robôs.

Classificação de Métodos de Programação



Programação por aprendizagem (*on-line*):

- Consiste em ensinar o robô guiando-o através da trajetória desejada pelo usuário.

Programação textual (*off-line*):

- Permite indicar a tarefa ao robô mediante o uso de uma linguagem de programação de alto nível.

Programação por aprendizagem

Passiva:

- O programador segura o extremo do braço do robô e o movimenta através da trajetória mais adequada;
- Aplicação típica:
 - ✓ Pintura.

Ativa:

- O programador usa uma caixa de aprendizagem (*teach-in box*) para controlar os motores das articulações ao longo da trajetória;
- Aplicações típicas:
 - ✓ Soldagem a ponto;
 - ✓ Paletização.

Programação por aprendizagem

Vantagens:

- útil e imprescindível em muitas ocasiões;
- fácil de aprender e utilizar.

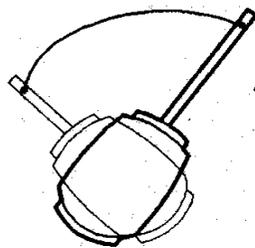
Desvantagens:

- necessidade de utilizar o próprio robô e seu ambiente de trabalho para realizar a programação, obrigando a interromper a linha de produção.
- cada movimento do operador é gravado e reproduzido de forma idêntica, incluindo movimentos não intencionais, a menos que haja mecanismo de edição.
- uma vez que a geração dos caminhos é feita manualmente, não é possível se obter alta precisão nem uma velocidade ótima para cada trajetória.
- dificuldade de integração de informação sensorial.

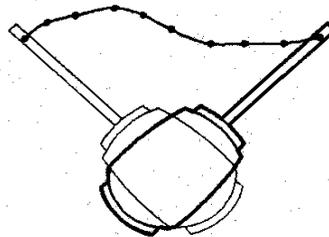
Programação por aprendizagem

Movimentos possíveis em programação por aprendizagem:

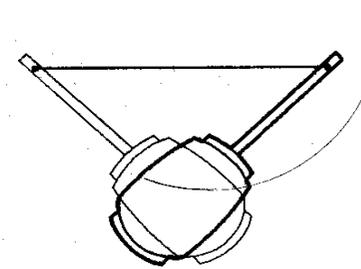
- **Ponto a ponto:** o robô se movimenta de uma posição inicial a outra final, sem que posições intermediárias sejam inseridas na programação da trajetória.
- **Movimento contínuo:** o robô se movimenta através de pontos com pequenos incrementos entre si, armazenados ao longo de uma trajetória previamente percorrida.
- **Controle de trajetória:** Envolve o controle coordenado de todas as juntas, para percorrer uma trajetória desejada ao longo de 2 pontos.



A) POINT-TO-POINT
PATH MOTION



B) CONTINUOUS PATH
MOTION



C) CONTROLLED PATH
MOTION

Fig. 19.1. Motion control algorithms determine the actual path taken when the program is replayed in the automatic mode.

Vantagens da programação textual (*off-line*)

- Redução do tempo ocioso: O robô pode ser mantido na linha de produção, enquanto a próxima tarefa está sendo programada.
- Operação mais segura: Redução no tempo de permanência do operador próximo ao robô.
- Simplificação de programação: Pode-se usar a forma *off-line* para programar uma grande variedade de robôs, sem a necessidade de se conhecerem as peculiaridades de cada controlador.
- Integração com sistemas CAD/CAM: habilita a interface com o banco de dados de peças, centralizando a programação de robôs com esses sistemas.
- Depuração de programas: Para sistemas integrados com CAD/CAM, ao se utilizar modelos para simular o comportamento real do robô pode-se detectar colisões dentro do espaço de trabalho evitando assim danos ao equipamento.

Limitações da programação textual (*off-line*)

- Dificuldade em se desenvolver um sistema de programação generalizado que seja independente do robô e de suas aplicações.
- Para reduzir a incompatibilidade entre robôs e sistemas de programação, faz-se necessário definir padrões para as interfaces.
- Programas gerados *off-line* devem levar em conta os erros e imprecisões entre o modelo idealizado e o mundo real.
- Modelo teórico e realidade não batem:
 - ✓ imprecisão na montagem dos componentes;
 - ✓ falta de rigidez na estrutura do robô;
 - ✓ resolução insuficiente do controlador;
 - ✓ imprecisão numérica;
 - ✓ dificuldade na determinação precisa dos objetos com relação ao sistema de coordenadas utilizado.

Classificação da Programação textual

Nível de Robô:

- As instruções do programa se referem aos movimentos a serem realizados pelo robô;
- É especificado cada um dos movimentos do robô incluindo velocidade, direções de aproximação e saída, abertura e fechamento da garra, etc;.
- É necessário decompor a tarefa global em várias sub-rotinas como movimentar o robô, agarrar os objetos, realizar uma inserção, etc.

Nível de Objeto:

- As instruções do programa se referem ao estado em que os objetos manipulados devem ficar;
- A complexidade do programa diminui;
- A programação se realiza de maneira mais conveniente, pois as instruções se realizam em função dos objetos a manipular.

Nível de Tarefa:

- As instruções do programa se referem ao objetivo a alcançar;
- O programa pode ser reduzido a uma única instrução.

Programação do robô KUKA do Laboratório

Programação por aprendizagem ativa:

- Usando *teach in box* ou *teach pendant*;
- Programação a nível de usuário;
- Movimentos possíveis de serem realizados ⇒ ponto a ponto, contínuo, controle de trajetória.

Programação textual a nível de robô:

- Linguagem de programação KRL;
- Programação a nível de especialista.

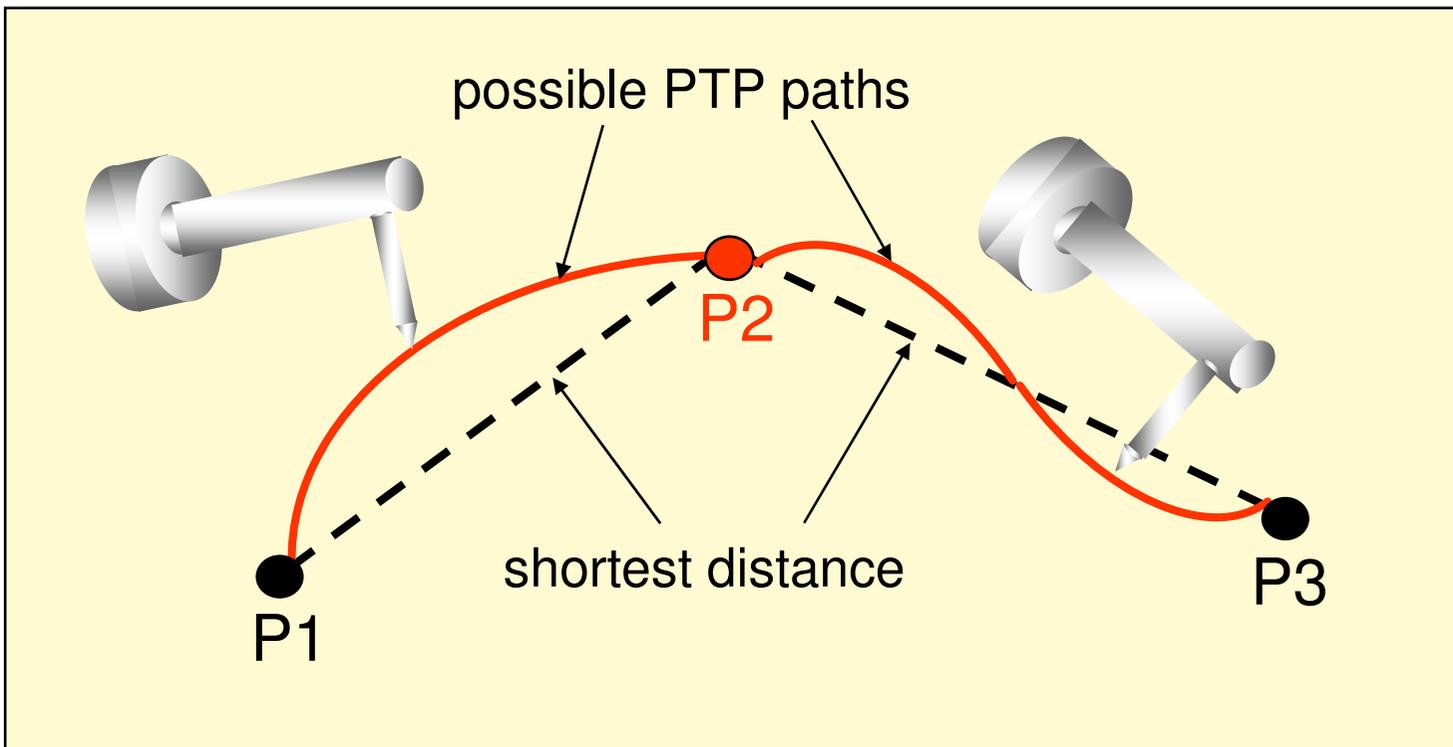
Programação de movimentos

- Abrir o menu usando a tecla de “COMMANDS”;
- Desse menu selecionar “MOTION”;
- Fazer a seleção da instrução de tipo de movimento (PTP, LIN or CIRC).

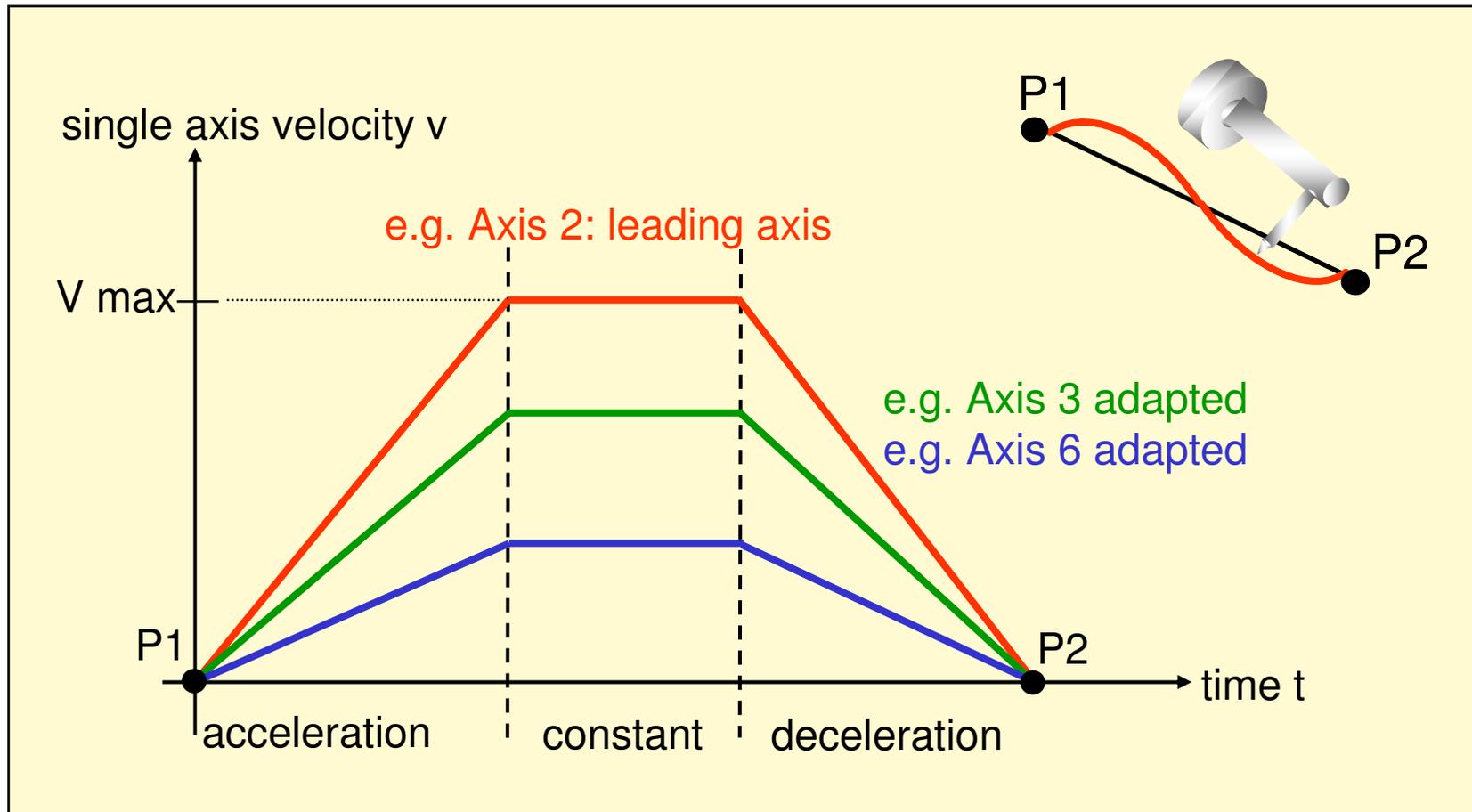


PTP motion with exact positioning

PTP motion without approximation → P2 exact position



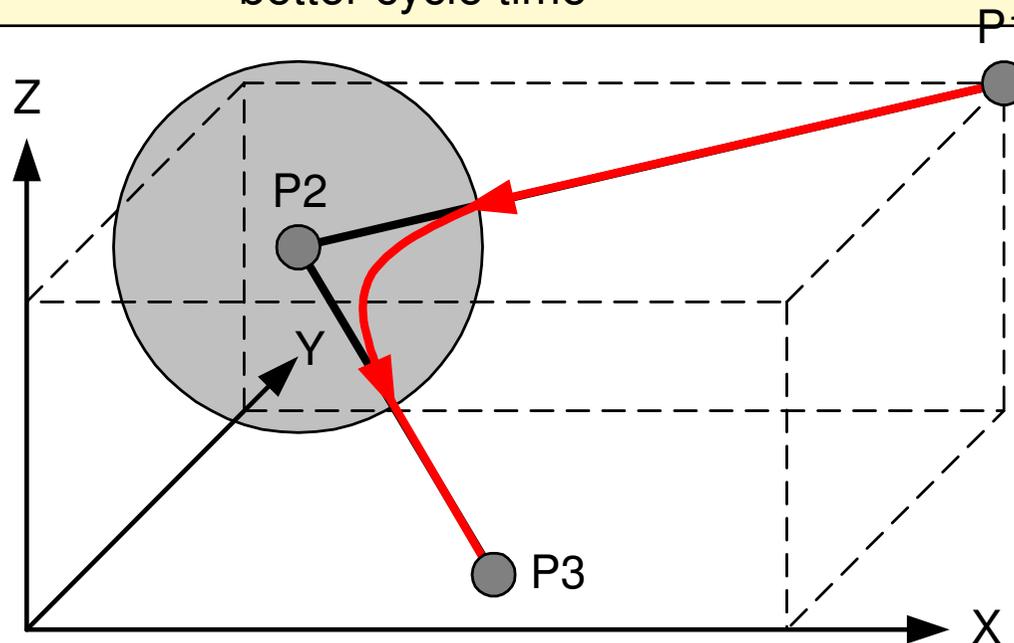
Motion profile (synchronized)



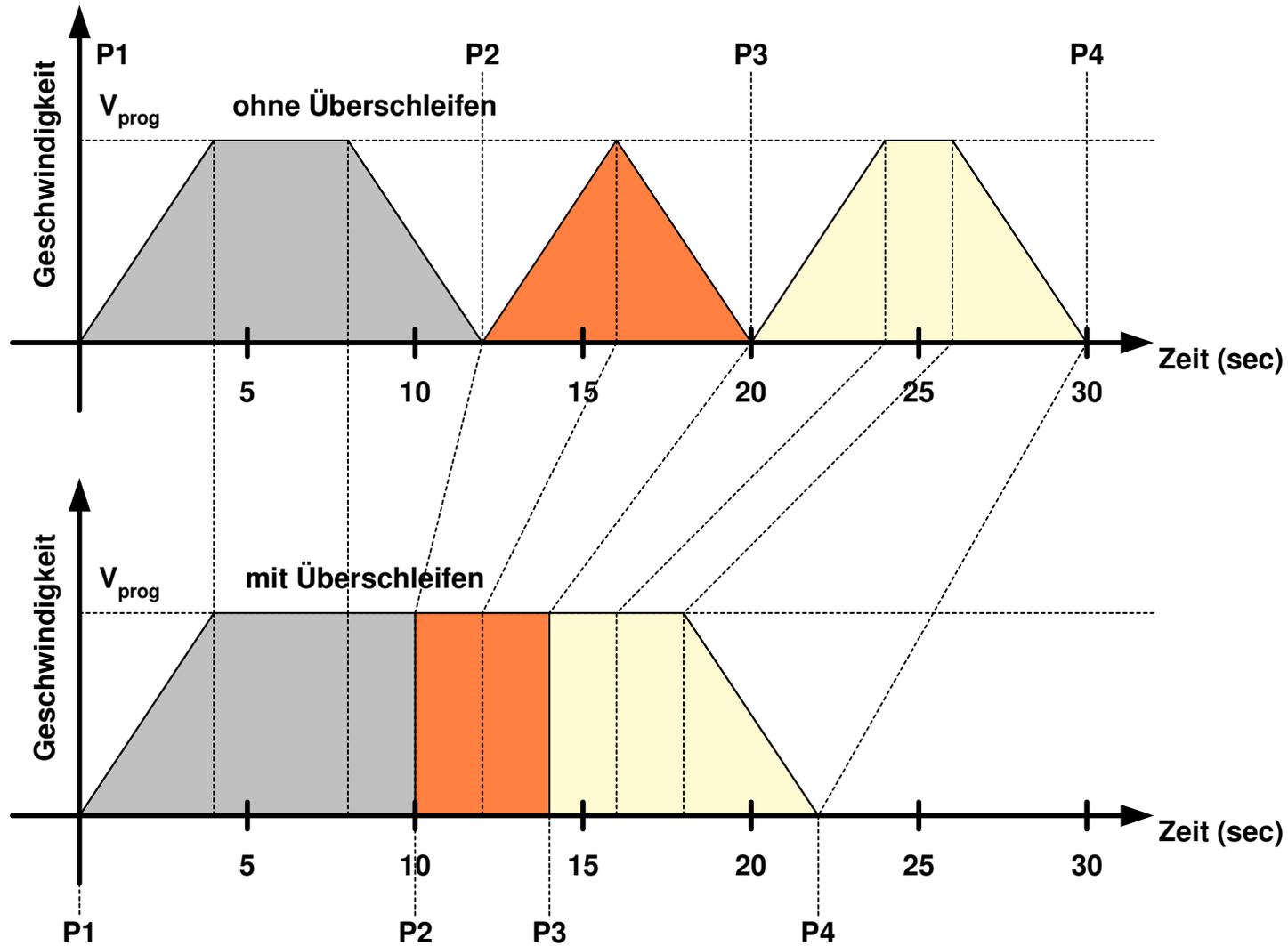
PTP-motion with approximate positioning

At a continuous path motion the destination position is not positioned exactly but approximated. Therefore the robot must not be decelerated to a stop. Advantages:

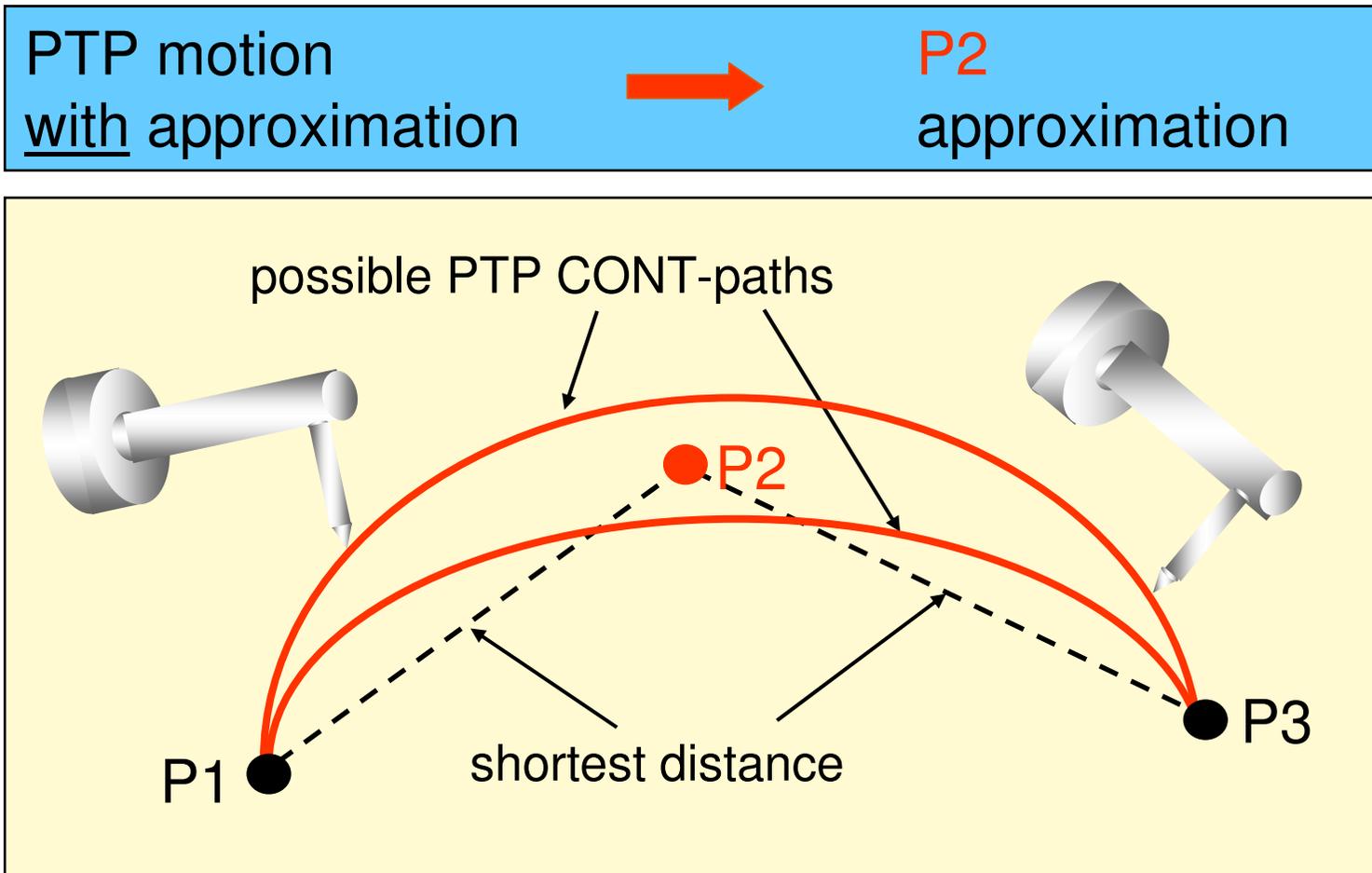
- less wear out
- better cycle time



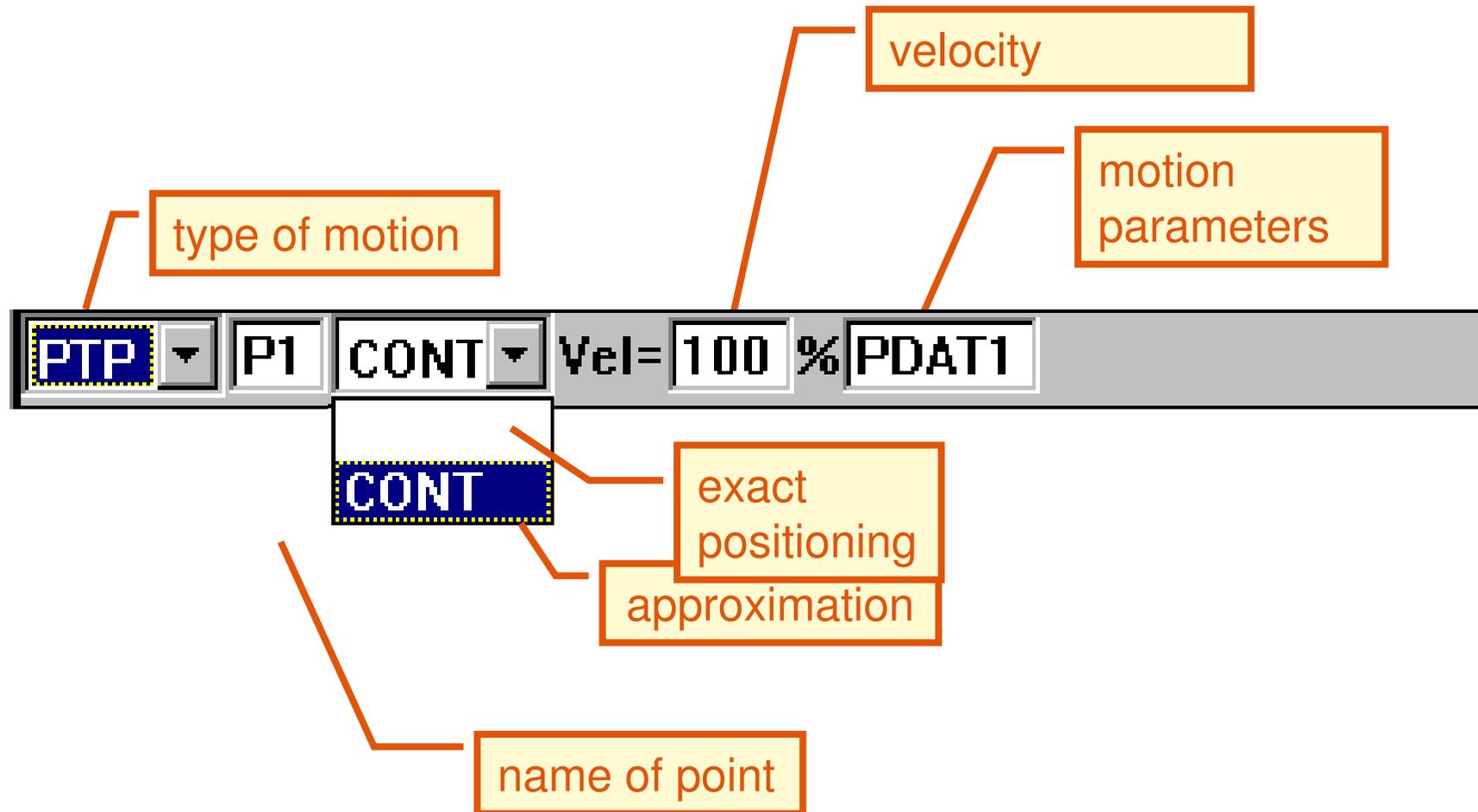
Improving the cycle time using continuous motion commands



PTP motion with approximate positioning



Programming a PTP motion



Programming a PTP motion

A vertical configuration panel with a cyan background. It has three sections: 'Tool' with a dropdown menu showing 'TOOL DATA[1]', 'Base' with a dropdown menu showing 'NULLFRAME', and 'external TCP' with a dropdown menu showing 'False'. Orange lines connect the dropdown menus to their respective callout boxes on the right.

Tool

select tool
Tool_Data[1]..[16], Nullframe

Base

select base
Base_Data[1]..[16], Nullframe

External TCP

Robot guiding tool: False
Robot guiding workpiece : True

Programming a PTP motion

The image shows a software interface for programming a PTP motion. At the top, a grey bar contains several controls: a dropdown menu set to 'PTP', a 'P1' label, another dropdown menu set to 'CONT', a velocity field 'Vel=' with '100' and a '%' symbol, and a parameter field 'PDAT2' which is circled in orange. Below the 'CONT' dropdown, a blue box labeled 'CONT' is connected by an orange line to a larger cyan box. This cyan box is titled 'Movement 1/1' and contains two settings: 'Acceleration' and 'Approximation Distance'. The 'Acceleration' setting has a value of '100' and a range from '1' to '100'. The 'Approximation Distance' setting has a value of '100' and a range from '0' to '100'. Two orange callout boxes are connected to the cyan box: one points to the 'Acceleration' field and the other points to the 'Approximation Distance' field.

Acceleration
To be used I the motion.
Value: 1..100%

Approximation Distance^{*)}
To be used in the motion.
Value: 0..100%

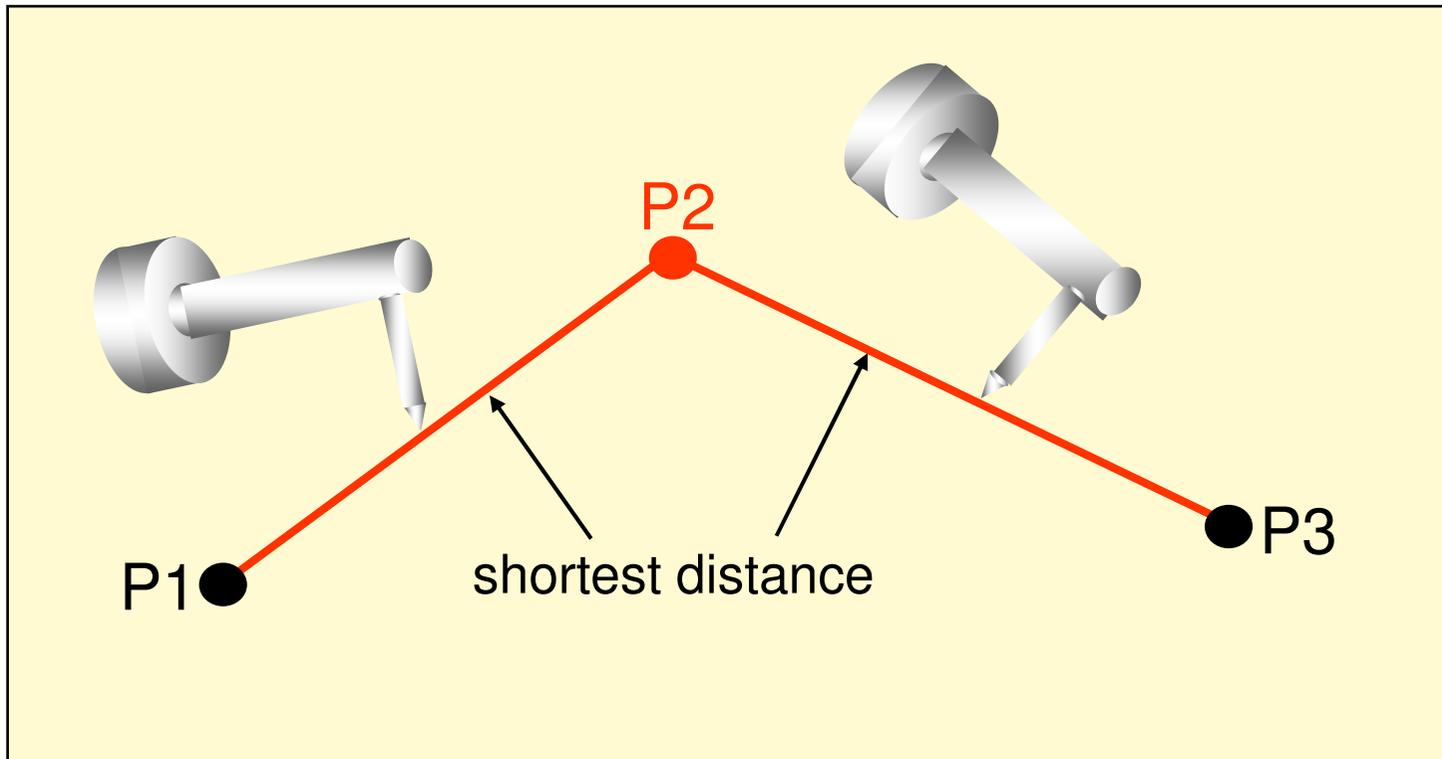
- *) The parameter „Approximation Distance“ is only available if “CONT” has been switched on.

LIN motion with exact positioning

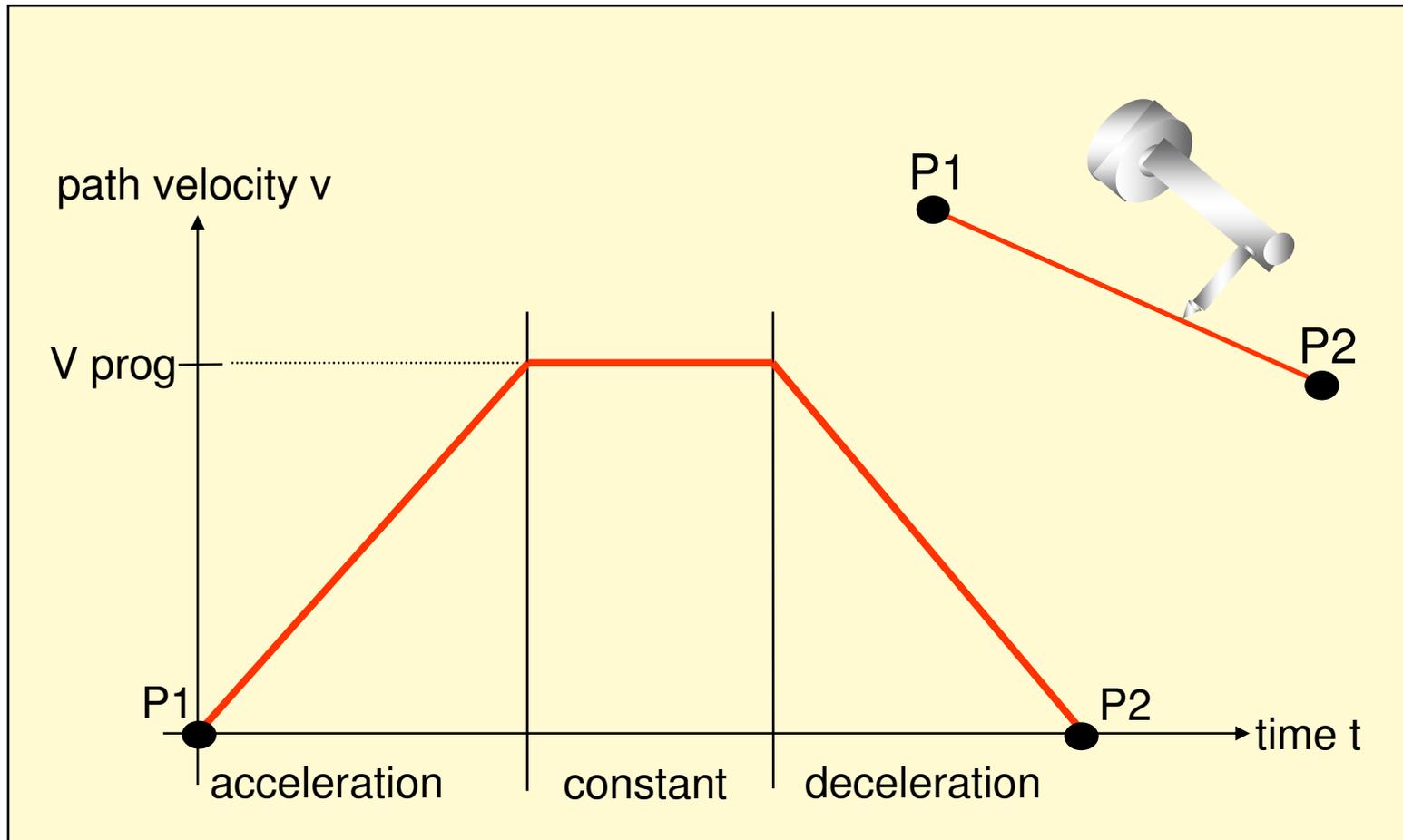
LIN motion
without approximation



P2
exact position

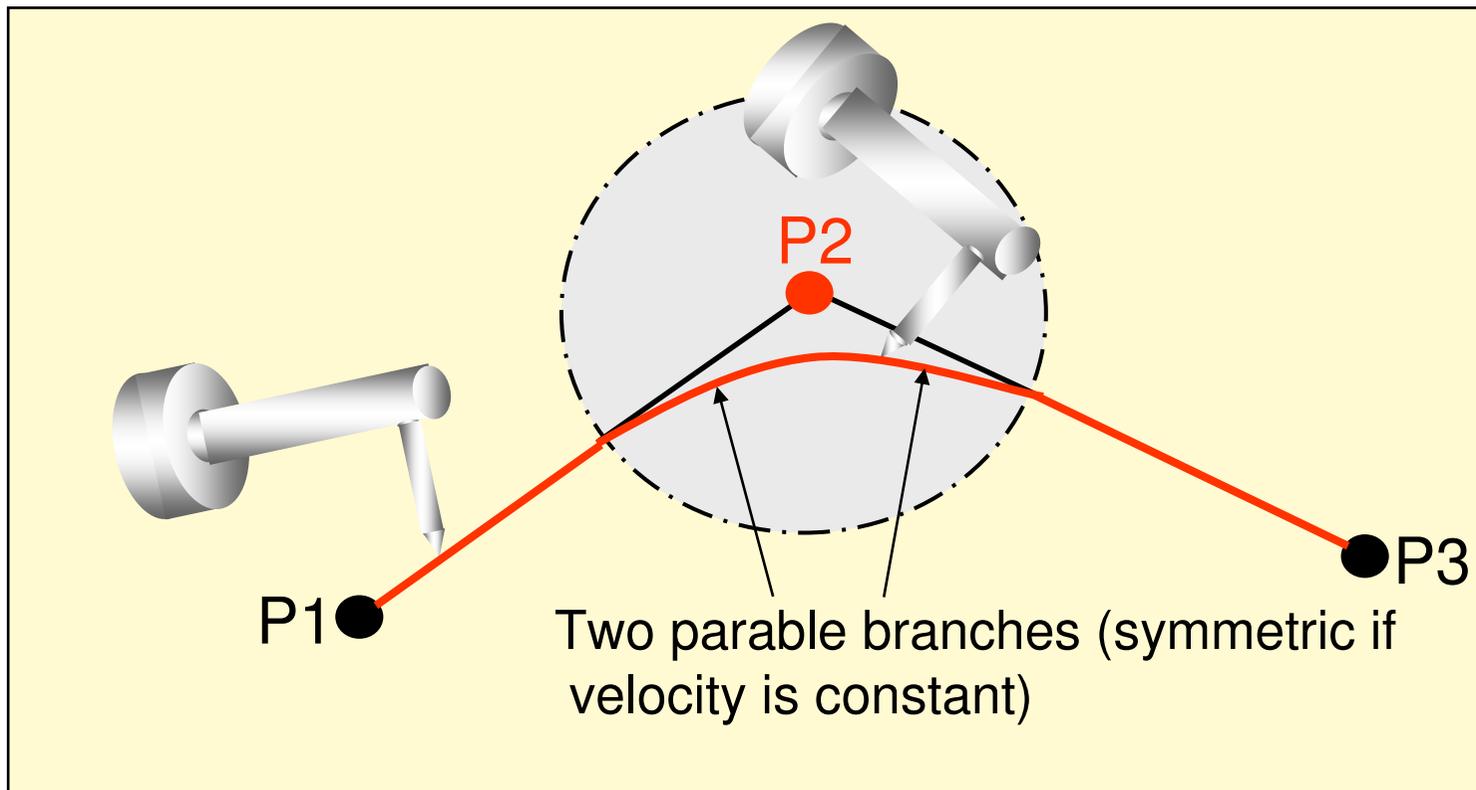


Velocity profile

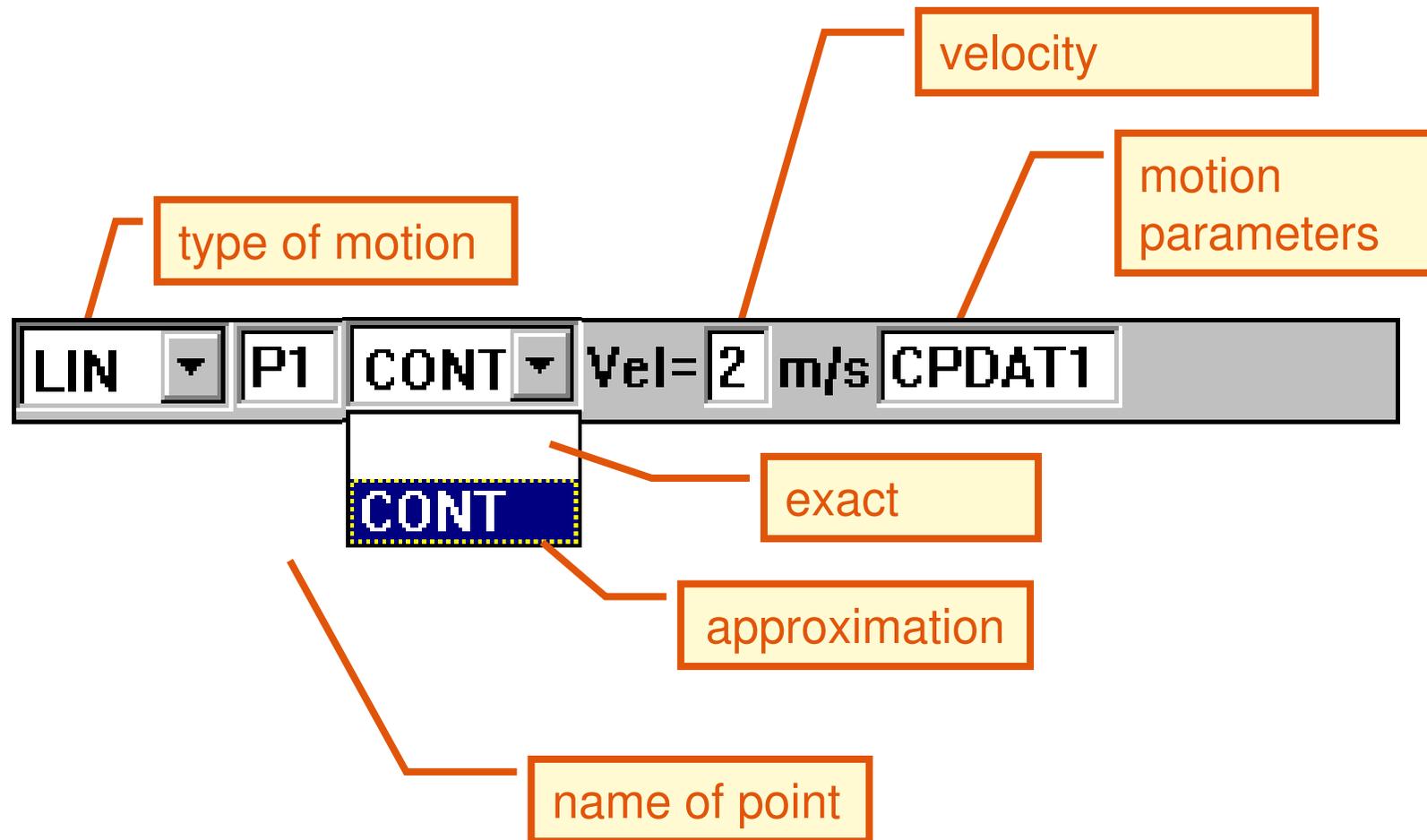


LIN motion with approximate positioning

LIN motion with approximation → P2 approximation



Programming a LIN motion



Programming a LIN motion



A screenshot of the 'Frames' configuration panel, which is highlighted in cyan. It contains three sections: 'Tool' with a dropdown menu showing 'TOOL_DATA[1]', 'Base' with a dropdown menu showing 'NULLFRAME', and 'external TCP' with a dropdown menu showing 'False'. Orange lines connect these dropdowns to their respective explanation boxes on the right.

Tool
select tool
Tool_Data[1]..[16], Nullframe

Base
select base
Base_Data[1]..[16], Nullframe

External TCP
Robot guiding tool: False
Robot guiding workpiece : True

Programming a LIN motion

The image shows a software interface for programming a linear (LIN) motion. At the top, a grey bar contains the following elements from left to right: a dropdown menu set to 'LIN', a dropdown menu set to 'P1', a dropdown menu set to 'CONT', a text field containing 'Vel= 2 m/s', and a blue button labeled 'CPDAT1' which is circled in orange. Below the 'CONT' dropdown, a blue button labeled 'CONT' is shown. An orange line connects this 'CONT' button to a larger cyan window titled 'Movement 1/1'. This window contains two parameter settings, each with a slider and a text input field. The first is 'Acceleration' with a value of '100 %' and a range from '1' to '100'. The second is 'Approximation Distance' with a value of '100 mm' and a range from '0' to '100'. Two orange callout boxes are connected to these settings. The first callout box, titled 'Acceleration', contains the text: 'Acceleration to be used in the motion. Value: 1..100%'. The second callout box, titled 'Approximation Distance*)', contains the text: 'Approximation Distance*) to be used in the motion. Value: 0..100mm'.

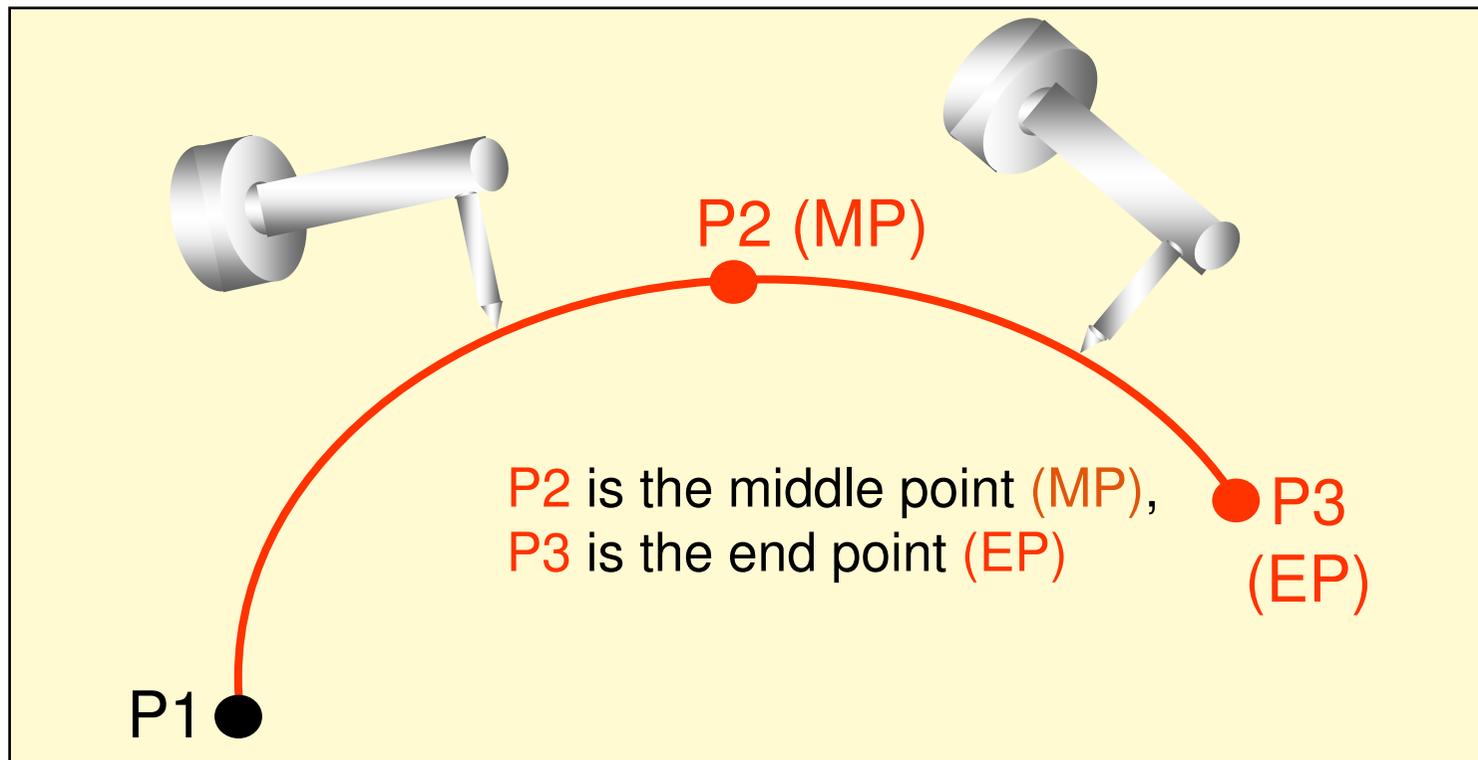
- *) The parameter „Approximation Distance“ is only available if “CONT” has been switched on.

CIRC motion with exact positioning

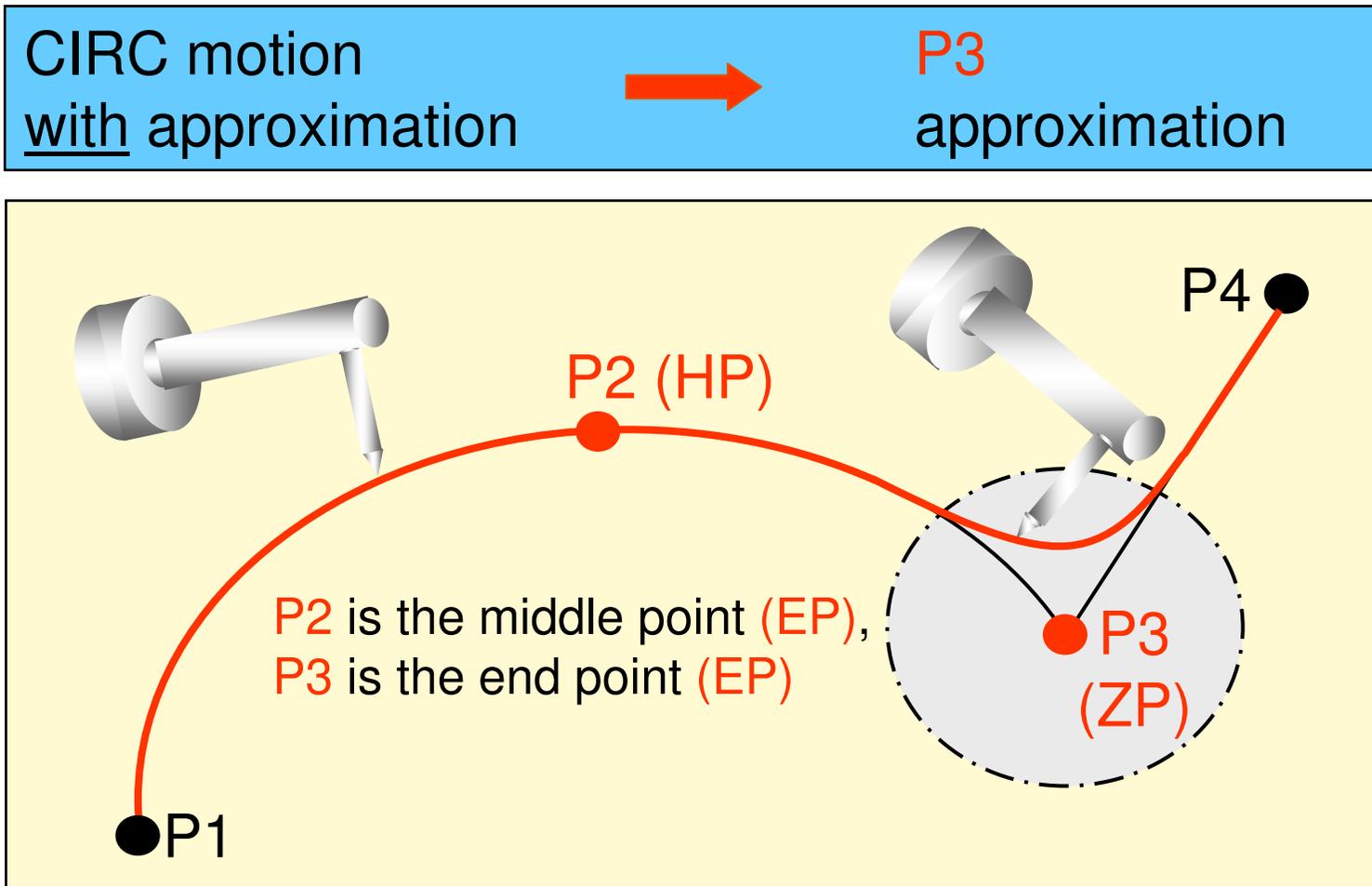
CIRC motion
without approximation



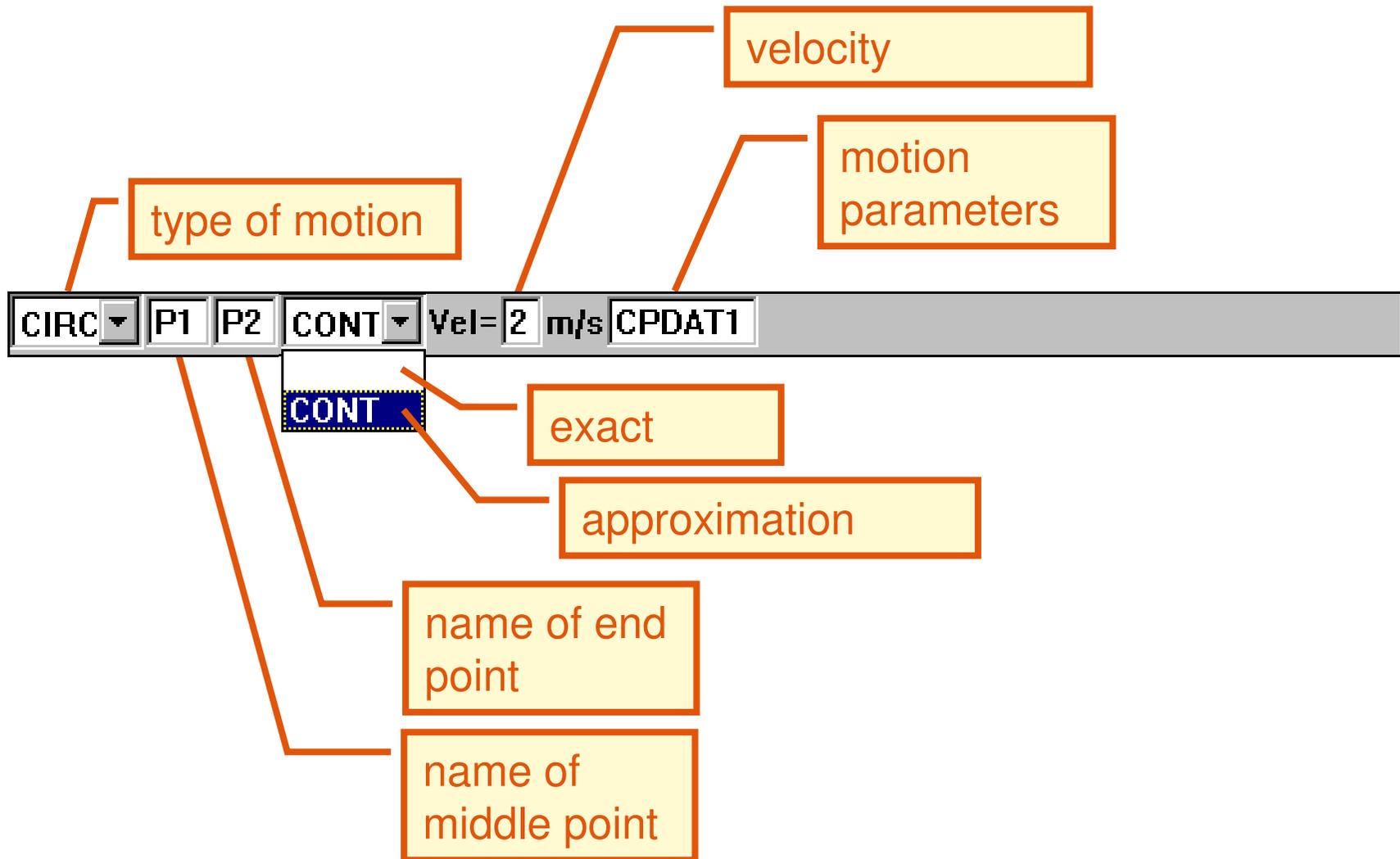
P3
exact position



CIRC motion with approximate positioning



Programming a CIRC motion



Programming a CIRC motion



A screenshot of the 'Frames' configuration panel. It has a cyan background and is titled 'Frames 1/1'. It contains three sections: 'Tool' with a dropdown menu showing 'TOOL_DATA[1]', 'Base' with a dropdown menu showing 'NULLFRAME', and 'external TCP' with a dropdown menu showing 'False'. Red lines connect these dropdowns to explanatory text boxes on the right.

Tool
select tool
Tool_Data[1]..[16], Nullframe

Base
select base
Base_Data[1]..[16], Nullframe

External TCP
Robot guiding tool: False
Robot guiding workpiece : True

Programming a CIRC motion

The image shows a control panel for a CIRC motion. At the top, there is a row of buttons: 'CIRC' (with a dropdown arrow), 'P1', 'P2', 'CONT' (with a dropdown arrow), 'Vel= 2 m/s', and 'CPDAT1' (circled in orange). Below the 'CONT' button is a 'CONT' label in a blue box. An orange line connects this 'CONT' label to a larger cyan box representing the motion configuration window. This window has a title bar 'Movement 1/1' and two main sections: 'Acceleration' and 'Approximation Distance'. The 'Acceleration' section has a slider from 1 to 100 with a value of 100. The 'Approximation Distance' section has a slider from 0 to 100 with a value of 100. Two orange callout boxes point to these sections. The first callout box, titled 'Acceleration', contains the text: 'Acceleration to be used in the motion. Value: 1..100%'. The second callout box, titled 'Approximation Distance*)', contains the text: 'Approximation Distance*) to be used in the motion. Value: 0..100mm'.

- *) The parameter „Approximation Distance“ is only available if “CONT” has been switched on.

The whole circle consist out of at least two different parts.

