FAST MOVING TECHNOLOGY

STÄUBLI

VAL3 Velocity addon

The velocity addon provides motion instructions to control the arm in velocity, rather than in position.
The use of these instructions should be safe (see potential bug with $setVelCmd), safety rules should be applied (monitor speed, move/Hold button, eStop, Cartesian limit in manual mode) but this has not been tested by R&D.

void **$velFrame**(frame fReference, tool tTool, mdesc mDesc):
Initializes a velocity move in the Cartesian space using a frame as reference.
The motion descriptor specifies the speed and acceleration limits to be applied during the move.
The tool is used for the Cartesian speed computations; the Cartesian speed limits of the motion descriptor are computed for this tool.

void **$velJoint**(tool tTool, mdesc mDesc):
Initializes a velocity move in the joint space
The motion descriptor specifies the speed and acceleration limits to be applied during the move.
The tool is used for the Cartesian speed computations; the Cartesian speed limits of the motion descriptor are computed for this tool.

void **$velTool**(tool, mdesc):
Initializes a velocity move in the Cartesian space using a tool as reference
The motion descriptor specifies the speed and acceleration limits to be applied during the move.
The tool is used for the Cartesian speed computations; the Cartesian speed limits of the motion descriptor are computed for this tool.
$velTool makes the system fail with RS arms in VAL3 -s6.3.

(s6.4+) void **$setVelCmd**(num cmd[6]):
Specify the speed for the 6 coordinates (mm/s (x, y, z) and deg/s (rx, ry, rz) for frame and tool modes, deg/s (rotoid axis) mm/s (linear axis) for joint mode)

void **$velCmd**(num cmd[6]) (to be removed with VAL3 s7)
Same as $setVelCmd, but specifies the speed in m/s or rad/s instead mm/s and deg/s.
$velCmd makes the system fail with RS arms in Frame mode in VAL3 -s6.3.

- Once the vel command is initialized, the $setVelCmd instruction can change the speed (and direction with negative speed) in real time. The change is applied by the synchronuous task every 4ms.

⚠ The $setVelCmd () is not real-time proof and a partial, possibly corrupted, command can result if the synchronuous task is activated during the execution of the VAL3 velCmd instruction (very rare). This can be avoided by the use of a synchronuous task.

- To stop the velocity command, a resetMotion() is required ($setVelCmd () with cmd = 0 will stop the arm, but the vel instruction is still activated).

A very simple example of use of the vel commands :

```
num nCmd[6]
begin
$velFrame(world,flange,mNominal)
while true
nCmd[0]=100
nCmd[1]=100
$setVelCmd (nCmd)
delay(3)
nCmd[0]=-100
nCmd[1]=-100
$setVelCmd (nCmd)
delay(3)
endWhile
end
```