

Programming Practices That Kill Cycle-Time ...and Options to Improve It

Introduction

Why is my job running so slow? How can the cycle-time be improved?

There are several reasons why your cycle-time may not be where it should be. It is important to note that the YRC1000 is naturally “path accurate”. That is, it *exactly tracks* the 6-degrees of freedom shape of the desired path in axis space. (The computed trajectory is as close as possible to the original robot program.) YRC1000 path accuracy can especially be seen when using linear interpolation; and rounding corners. In addition, there can be other factors at play leading to longer cycle times like:

- no tool data
- slow accelerating/decelerating or
- having improperly placed timer commands.

While there can be many factors leading to unexpectedly long cycle times, this document discusses some common causes and ways to address them.

Common Practices that Cause Slow Cycle-Time

Tool Data

This can be the most common culprit. If the tool data is missing, then the manipulator *assumes the heaviest load it can carry*...slowing the cycle-time. Always make sure the tool data is filled out.

TOOL					
TOOL NO. : 0					
NAME	STANDARD TOOL				
X	0.000	mm	Rx	0.0000	deg.
Y	0.000	mm	Ry	0.0000	deg.
Z	0.000	mm	Rz	0.0000	deg.
W	0.000	kg			
Xg	0.000	mm	Ix	0.000	kg. m2
Yg	0.000	mm	Iy	0.000	kg. m2
Zg	0.000	mm	Iz	0.000	kg. m2

Old Controller Job NOT Optimized

This is a less common issue. However, it is something to look for when converting a job from an older controller (NX100, DX100, DX200) to YRC1000. Older controllers were not as path accurate as the YRC1000, so there may be code:

- ACC/DEC tags
- PL tags
- Timers
- Extra Taught Points

Check the job for this kind of code or the cycle-time will be slower than the old controller.

Long Timers

Be careful when using timers in a job. They should be used sparingly. When using a timer (TIMER T=1) try to keep the time (in seconds) as low as possible (.1 sec) when using to improve path accuracy.

When using a timer in simulation (gripper actuation time or vision acquisition) verify the estimated times with your customer.

PL Statements

Check for PL=0 or PL=1 in the job. They might be added to force the manipulator to stop at an intermediate point. It might be best to use a small timer instead (.1 sec). PL=0 causes the manipulator to decelerate more...extending the time it takes to complete the job.

Nested Jobs & Math Functions

Nested jobs and Math Functions may slow the manipulator down but only if the cycle-time is very small. An example would be if the required cycle-time is 1.5 sec, but the job results in a 3 sec cycle-time. The controller is always calculating several steps ahead. Nested jobs and math functions create more calculations than hard-taught points...resulting in a slightly slower cycle-time.

Wrong Robot

This is hopefully never the cause, but it is critical that the right manipulator is selected prior to installation. If a manipulator is too slow the only real option is either change the cell layout or change the manipulator. For example, a customer wants to use an HC20XP COBOT to palletize 40 cases per minute. A COBOT is not as fast as a true palletizing manipulator like the PL190. ALWAYS choose the right manipulator model for the task.

This document captures ideas, experiences, and informal recommendations from the Yaskawa Partner Support team. It is meant to augment – not supersede manuals or documentation from motoman.com. Please contact the Partner Support team at partnersupport@motoman.com for updates or clarification.

Simple Techniques that Can Improve Cycle-Time

VMAX

VMAX is a tag...added to a linear move command, MOVL. It can push the manipulator to go beyond its max linear speed and improve cycle-time sometimes as much as 30%.

NOTE: This command is only recommended when path accuracy is not critical. VMAX is only available on the YRC1000

Joint Move Commands

When programming large moves between locations, especially if the primary axis in use is the S-axis...try using a MOVJ (joint command) instead of a MOVL (linear command). Joint moves allow the manipulator to decide the best path to take without constraint.

Speed Priority Control Function

When the distance between teaching positions is short, the manipulator may move at the slower speed than the specified speed because the required distance is not enough to reach the specified speed. Using the Speed Priority Control Function the manipulator moves at the specified speed even when the distance between teaching positions is short.

Conclusion

There are many ways a job can run too slow and there are many ways to speed that job up. For more details see the YRC1000 General Operator's Manual, or the Operator's Manual for your specific controller on motoman.com