

Cobot SDK user manual version update content(V3.8.0)

**Revision Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Version** | **Description** |
| 2025/01/08 | V3.7.7 | Create |
| 2025/01/20 | V3.8.0 | 1. EndForceDragControl() added params for ingularity Constraints  2.ArcWeldTraceControl() added params for offset  3.add WeaveChangeStart() command  4.add WeaveChangeEnd()command  5.add LoadTrajectoryLA()command  6.add MoveTrajectoryLA()command  7.add CustomCollisionDetectionStart()command  8.add CustomCollisionDetectionEnd()command |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**Notes:**

**①This manual is applicable to WebApp version 3.8.0 Cobots and is subject to change without notice. For other versions , please visit：**

[Welcome to FAIR’s documentation! — FAIRINO Collaborative Robot User Manual 3.7.8 documentation](https://fair-documentation.readthedocs.io/en/latest/)

Catalogue

[1 Modify the interface 1](#_Toc190)

[1.1 Force sensor assists drag 1](#_Toc9984)

[1.1.1 Interface Description 1](#_Toc15644)

[1.1.2 Example 1](#_Toc3921)

[1.2 Arc tracking control 2](#_Toc3462)

[1.2.1 Interface Description 2](#_Toc23839)

[1.2.2 Example 3](#_Toc21914)

[2 Added Command 4](#_Toc9349)

[2.1 Wobble gradient begins and ends 4](#_Toc1049)

[2.1.1 Interface Description 4](#_Toc24593)

[2.1.2 Example 4](#_Toc31998)

[2.2 trajectory Preprocessing (trajectory Foresight)、trajectory reproduction(trajectory Foresight) 5](#_Toc28903)

[2.2.1 Interface Description 5](#_Toc23950)

[2.2.2 Example 6](#_Toc1315)

[2.3 custom collision detection threshold function starts ends 7](#_Toc12725)

[2.3.1 Interface Description 7](#_Toc15932)

[Section: 6 Security settings 7](#_Toc11619)

[2.3.2 Example 8](#_Toc32618)

# Modify the interface

## Force sensor assists drag

### Interface Description

Section: 11.25. Force sensor assists drag

|  |
| --- |
| 1. /\*\* 2. \* @brief Force sensor assisted drag 3. \* @param [in] status control status, 0-off; 1-on 4. \* @param [in] asaptiveFlag Adaptive on flag, 0-off; 1-on 5. \* @param [in] interfereDragFlag Interference area drag flag, 0-off; 1-on 6. \* @param [in] ingularityConstraintsFlag singularity strategy, 0-avoidance; 1-crossing 7. \* @param [in] M inertia factor 8. \* @param [in] B damping factor 9. \* @param [in] K Stiffness factor 10. \* @param [in] F Dragging six-dimensional force threshold 11. \* @param [in] Fmax Maximum drag force limit Nm 12. \* @param [in] Vmax Maximum joint speed limit °/s 13. \* @return Error code 14. \*/ 15. int EndForceDragControl(int status, int asaptiveFlag, int interfereDragFlag, int ingularityConstraintsFlag,double[] M, double[] B, double[] K, double[] F, double Fmax, double Vmax); |

## Arc tracking control

### Interface Description

Section: 12.23. Arc tracking control

|  |
| --- |
| 1. / \* \* 2. \* @brief arc tracking control 3. \* @param [in] flag Switch, 0- off. 1-on 4. \* @param [in] dalayTime Delay time (unit: ms) 5. \* @param [in] isLeftRight left and right deviation compensation 6. \* @param [in] klr adjustment coefficient (sensitivity) 7. \* @param [in] tStartLr or around start compensation time cyc 8. \* @param [in] stepMaxLr About the maximum compensation amount mm each time 9. \* @param [in] sumMaxLr Maximum total compensation mm 10. \* @param [in] isUpLow up-and-down deviation compensation 11. \* @param [in] kud up-down adjustment factor (sensitivity) 12. \* @param [in] tStartUd start up and down compensation time cyc 13. \* @param [in] stepMaxUd Maximum compensation amount mm each time 14. \* @param [in] sumMaxUd Maximum total compensation 15. \* @param [in] axisSelect upper and lower coordinate system selection, 0-swing; 1- Tools; 2- Base 16. \* @param [in] referenceType Reference current setting mode, 0- feedback; 1-constant 17. \* @param [in] referSampleStartUd Reference current sampling starts counting (feedback), cyc 18. \* @param [in] referSampleCountUd Reference current sampling cycle count (feedback), cyc 19. \* @param [in] referenceCurrent Upper and lower reference current mA 20. \* @param [in] offsetType Indicates the offset tracking type. 0- no offset. 1- Sampling; 2- percentage 21. \* @param [in] offsetParameter Offset parameter; Sampling (offset sampling start time, default sampling cycle); Percentage (offset percentage (-100 to 100)) /version 3.7.9 22. \* @return Error code 23. \* / 24. int ArcWeldTraceControl(int flag, double delaytime, int isLeftRight, double klr, double tStartLr, double stepMaxLr, double sumMaxLr, int isUpLow, double kud, double tStartUd, double stepMaxUd, double sumMaxUd, int axisSelect, int referenceType, double referSampleStartUd, double referSampleCountUd, double referenceCurrent, int offsetType, int offsetParameter); |

### Example

Section：11.62

|  |
| --- |
| 1. private void btnweld\_Click(object sender, EventArgs e) 2. { 3. //电弧跟踪 4. DescPose p1Desc = new DescPose(-72.912, -587.664, 31.849, 43.283, -6.731, 15.068); 5. JointPos p1Joint = new JointPos(74.620, -80.903, 94.608, -109.882, -90.436, -13.432); 6. DescPose p2Desc = new DescPose(-104.915, -483.712, -25.231, 42.228, -6.572, 18.433); 7. JointPos p2Joint = new JointPos(66.431, -92.875, 116.362, -120.516, -88.627, -24.731); 8. DescPose p3Desc = new DescPose(-242.834, -498.697, -23.681, 46.576, -5.286, 8.318); 9. JointPos p3Joint = new JointPos(57.153, -82.046, 104.060, -116.659, -92.478, -24.735); 10. ExaxisPos exaxisPos = new ExaxisPos(0.0, 0.0, 0.0, 0.0); 11. DescPose offdese = new DescPose(0.0, 0.0, 0.0, 0.0, 0.0, 0.0); 12. robot.WeldingSetVoltage(1, 19, 0, 0); 13. robot.WeldingSetCurrent(1, 190, 0, 0); 14. robot.MoveJ(p1Joint, p1Desc, 1, 1, 100, 100, 100, exaxisPos, -1, 0, offdese); 15. robot.MoveL(p2Joint, p2Desc, 1, 1, 100, 100, 50, -1, exaxisPos, 0, 0, offdese); 16. robot.ARCStart(1, 0, 10000); 17. robot.ArcWeldTraceControl(1, 0, 1, 0.06, 5, 5, 60, 1, 0.06, 5, 5, 60, 0, 0, 4, 1, 10, 2, 2); 18. robot.WeaveStart(0); 19. robot.MoveL(p3Joint, p3Desc, 1, 1, 100, 100, 1, -1, exaxisPos, 0, 0, offdese); 20. robot.WeaveEnd(0); 21. robot.ArcWeldTraceControl(0, 0, 1, 0.06, 5, 5, 60, 1, 0.06, 5, 5, 60, 0, 0, 4, 1, 10, 2, 2); 22. robot.ARCEnd(1, 0, 10000); 23. } |

# Added Command

## Wobble gradient begins and ends

### Interface Description

Section: 12 Robot Welding

|  |
| --- |
| 1. / \* \* 2. \* @brief swing gradient begins 3. \* @param [in] weaveNum Swing number 4. \* @return Error code 5. \* / 6. int WeaveChangeStart(int weaveNum) 7. / \* \* 8. \* @brief swing gradient ends 9. \* @return Error code 10. \* / 11. int WeaveChangeEnd() |

### Example

|  |
| --- |
| 1. private void btnweld\_Click(object sender, EventArgs e) 2. { 3. //摆动渐变 4. DescPose p1Desc = new DescPose(-72.912, -587.664, 31.849, 43.283, -6.731, 15.068); 5. JointPos p1Joint = new JointPos(74.620, -80.903, 94.608, -109.882, -90.436, -13.432); 6. DescPose p2Desc = new DescPose(-104.915, -483.712, -25.231, 42.228, -6.572, 18.433); 7. JointPos p2Joint = new JointPos(66.431, -92.875, 116.362, -120.516, -88.627, -24.731); 8. DescPose p3Desc = new DescPose(-240.651, -483.840, -7.161, 46.577, -5.286, 8.318); 9. JointPos p3Joint = new JointPos(56.457, -84.796, 104.618, -114.497, -92.422, -25.430); 10. ExaxisPos exaxisPos = new ExaxisPos(0.0, 0.0, 0.0, 0.0); 11. DescPose offdese = new DescPose(0.0, 0.0, 0.0, 0.0, 0.0, 0.0); 12. robot.WeldingSetVoltage(1, 19, 0, 0); 13. robot.WeldingSetCurrent(1, 190, 0, 0); 14. robot.MoveJ(p1Joint, p1Desc, 1, 1, 100, 100, 100, exaxisPos, -1, 0, offdese); 15. robot.MoveL(p2Joint, p2Desc, 1, 1, 100, 100, 50, -1, exaxisPos, 0, 0, offdese); 16. robot.ARCStart(1, 0, 10000); 17. robot.ArcWeldTraceControl(1, 0, 1, 0.06, 5, 5, 60, 1, 0.06, 5, 5, 80, 0, 0, 4, 1, 10, 0, 0); 18. robot.WeaveStart(0); 19. robot.WeaveChangeStart(1); 20. robot.MoveL(p3Joint, p3Desc, 1, 1, 100, 100, 1, -1, exaxisPos, 0, 0, offdese); 21. robot.WeaveChangeEnd(); 22. robot.WeaveEnd(0); 23. robot.ArcWeldTraceControl(0, 0, 1, 0.06, 5, 5, 60, 1, 0.06, 5, 5, 80, 0, 0, 4, 1, 10, 0, 0); 24. robot.ARCEnd(1, 0, 10000); 25. } |

## trajectory Preprocessing (trajectory Foresight)、trajectory reproduction(trajectory Foresight)

### Interface Description

Section: [8. Trajectory recurrence](https://fairino-doc-en.readthedocs.io/latest/SDKManual/CPPRobotTrajectoryRecurrence.html)

|  |
| --- |
| 1. / \* \* 2. \* @brief trajectory Preprocessing (trajectory Foresight) 3. \* @param [in] name Indicates the track file name 4. \* @param [in] mode Sampling mode. 0- Sampling is not performed. 1- equal data interval sampling; 2- Equal error limit sampling 5. \* @param [in] errorLim Error limit, using line fitting takes effect 6. \* @param [in] type Indicates the smoothing mode, 0-Bessel smoothing 7. \* @param [in] precision Smoothing precision. This parameter takes effect when Bezier smoothing is used 8. \* @param [in] vamx set maximum speed, mm/s 9. \* @param [in] Maximum acceleration set by amax, mm/s2 10. \* @param [in] Max acceleration set by jmax, mm/s3 11. \* @return Error code 12. \* / 13. int LoadTrajectoryLA(string name, int mode, double errorLim, int type, double precision, double vamx, double amax, double jmax); 14. / \* \* 15. \* @brief track reproduction (track outlook) 16. \* @return Error code 17. \* / 18. int MoveTrajectoryLA(); |

### Example

|  |
| --- |
| 1. private void button8\_Click(object sender, EventArgs e) 2. { 3. int rtn = 0; 4. string nameA = "/fruser/traj/A.txt"; 5. string nameB = "/fruser/traj/B.txt"; 6. rtn = robot.LoadTrajectoryLA(nameB, 0, 0, 0, 1, 100.0, 100.0, 1000.0);// 直线拟合 7. Console.WriteLine($"LoadTrajectoryLA rtn is {rtn}"); 8. DescPose startPos = new DescPose(0, 0, 0, 0, 0, 0); 9. robot.GetTrajectoryStartPose(nameA, ref startPos); 10. // 11. robot.MoveCart(startPos, 1, 0, (float)100.0, (float)100.0, (float)100.0, -1, -1); 12. rtn = robot.MoveTrajectoryLA(); 13. Console.WriteLine($"MoveTrajectoryLA rtn is {rtn}"); 14. } |

## custom collision detection threshold function starts ends

### Interface Description

Section: 6 [Security settings](https://fairino-doc-en.readthedocs.io/latest/SDKManual/CPPRobotSecuritySettings.html)

|  |
| --- |
| 1. /\*\* 2. \* @brief Customize the collision detection threshold function to start, set the collision detection threshold on the joint side and the TCP side 3. \* @param [in] flag 1-joint detection only on; 2-TCP detection only on; 3-joint and TCP detection both on 4. \* @param [in] jointDetectionThreshould Joint collision detection threshold j1-j6 5. \* @param [in] tcpDetectionThreshould TCP collision detection threshold, xyzabc 6. \* @param [in] block 0-non-blocking; 1-blocking 7. \* @return Error code 8. \*/ 9. int CustomCollisionDetectionStart(int flag, double[] jointDetectionThreshould, double[] tcpDetectionThreshould, int block); 10. /\*\* 11. \* @brief Customize collision detection threshold function off 12. \* @return Error code 13. \*/ 14. int CustomCollisionDetectionEnd() |

### Example

|  |
| --- |
| 1. private void btnRobotSafetySet\_Click(object sender, EventArgs e) 2. { 3. while (true) 4. { 5. int[] safety = { 5, 5, 5, 5, 5, 5 }; 6. robot.SetCollisionStrategy(3, 1000, 150, 250, safety); 7. double[] jointDetectionThreshold = { 0.3, 0.3, 0.3, 0.3, 0.3, 0.3 }; 8. double[] tcpDetectionThreshold = { 80, 80, 80, 80, 80, 80 }; 9. int rtn = robot.CustomCollisionDetectionStart(3, jointDetectionThreshold, tcpDetectionThreshold, 0); 10. Console.WriteLine($"CustomCollisionDetectionStart rtn is {rtn}"); 11. DescPose p1Desc = new DescPose(228.879, -503.594, 453.984, -175.580, 8.293, 171.267); 12. JointPos p1Joint = new JointPos(102.700, -85.333, 90.518, -102.365, -83.932, 22.134); 13. DescPose p2Desc = new DescPose(-333.302, -435.580, 449.866, -174.997, 2.017, 109.815); 14. JointPos p2Joint = new JointPos(41.862, -85.333, 90.526, -100.587, -90.014, 22.135); 15. ExaxisPos exaxisPos = new ExaxisPos(0.0, 0.0, 0.0, 0.0); 16. DescPose offdese = new DescPose(0.0, 0.0, 0.0, 0.0, 0.0, 0.0); 17. // 假设MoveL方法签名如下： 18. robot.MoveL(p1Joint, p1Desc, 0, 0, 100, 100, 100, -1, exaxisPos, 0, 0, offdese); 19. robot.MoveL(p2Joint, p2Desc, 0, 0, 100, 100, 100, -1, exaxisPos, 0, 0, offdese); 20. rtn = robot.CustomCollisionDetectionEnd(); 21. Console.WriteLine($"CustomCollisionDetectionEnd rtn is {rtn}"); 22. } 23. } |