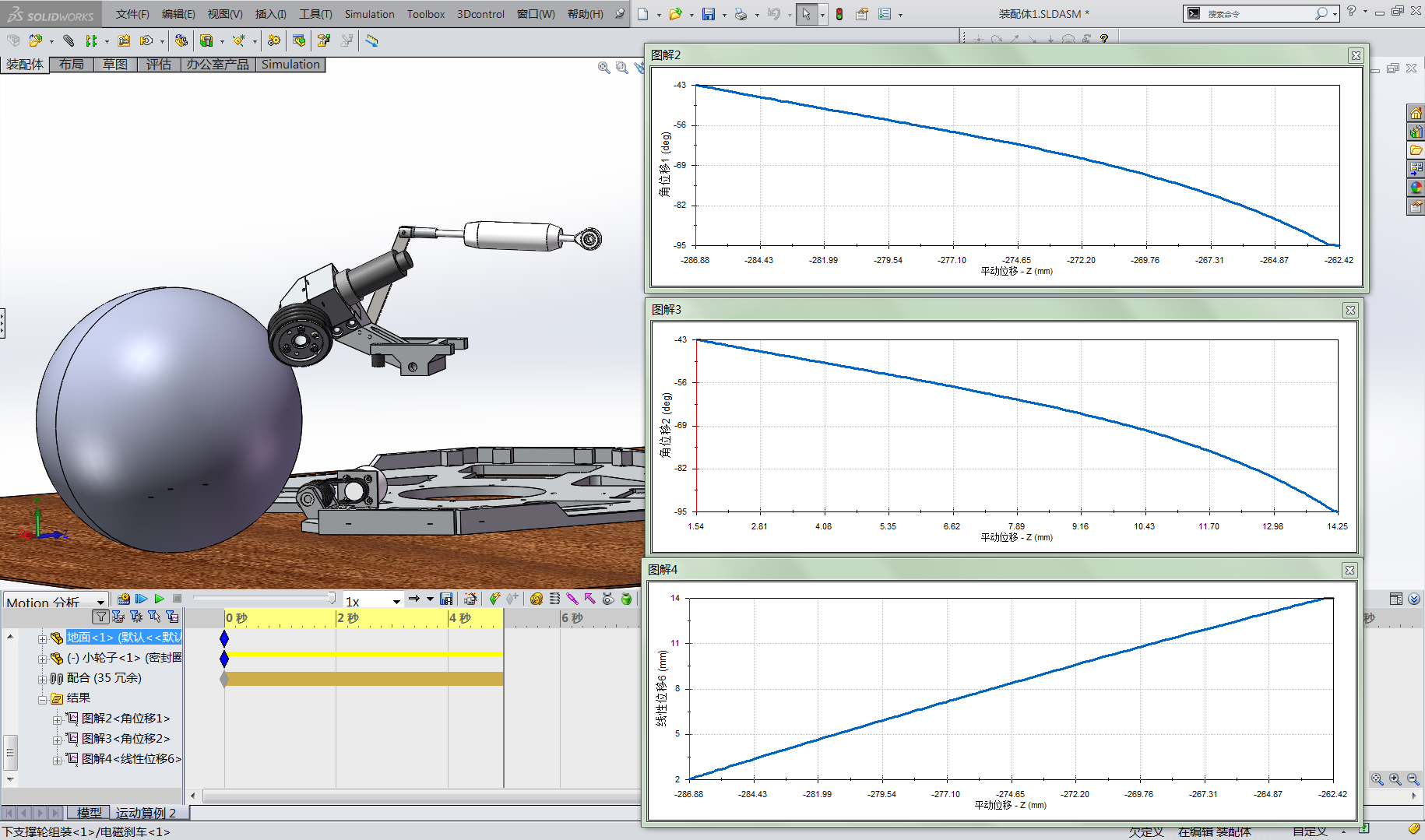
主动带球控制研究

利用Solidwork的motion analysis功能对主动带球进行运动仿真，可以得出球从完全脱离机器人（主动轮位于最低点）到完全进入机器人而抵住被动导轮整个过程中，球离机器人的距离(24-0mm)、直线位移传感器的伸缩距离(13-0mm)、主动轮的转动角度(0-62度)三者的变化关系：

1. 球距与主动轮转角的非线性关系，图解2；
2. 直线位移传感器的伸缩距离与主动轮转角的非线性关系，图解3；
3. 球距与直线位移传感器的伸缩距离的线性关系，图解4。

搞清楚这三者的变化关系有助于提高主动带球控制算法的性能。



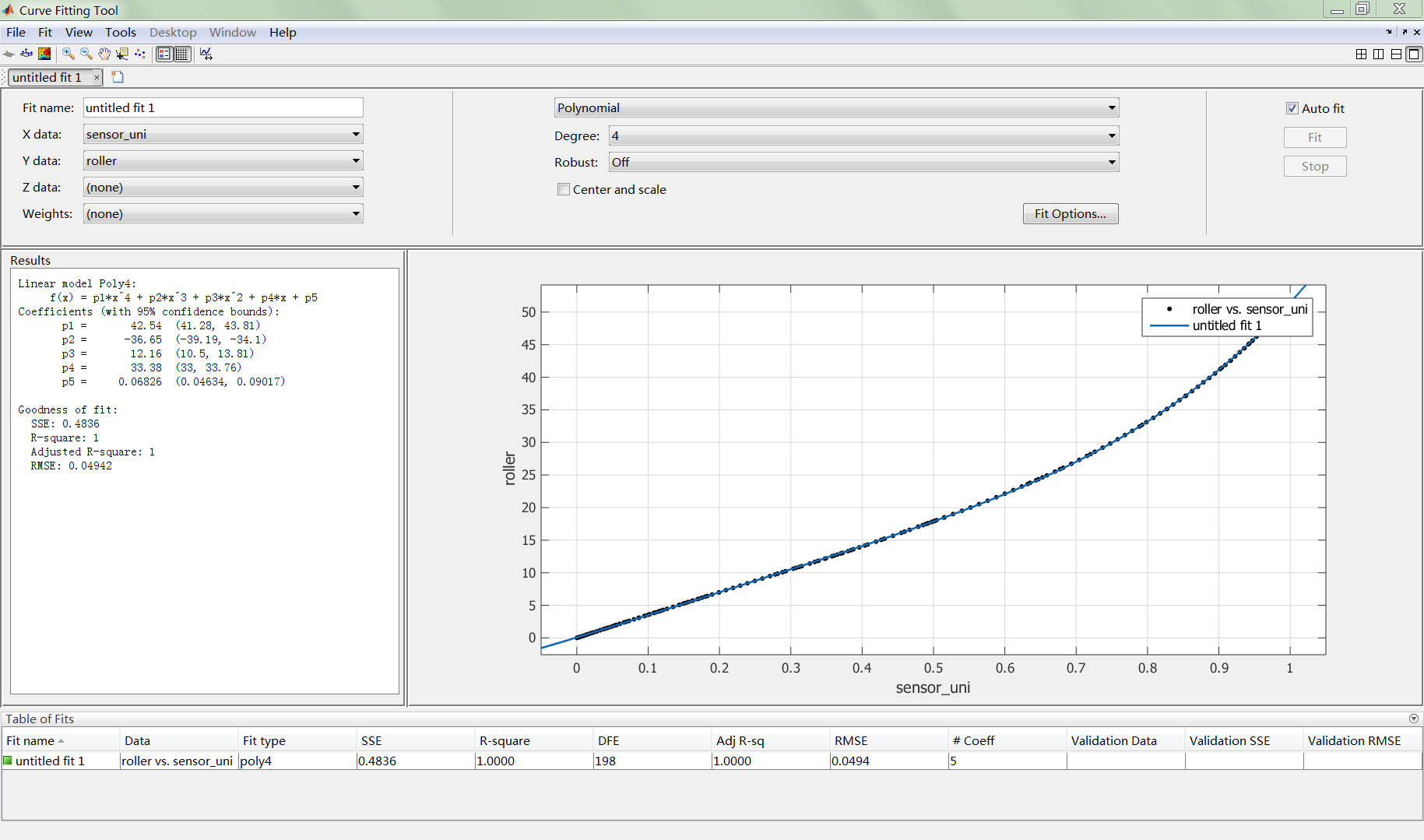
1. 直线位移传感器读数与主动轮转角

以前的主动球控制算法都是将归一化以后的传感器读数作为参考目标，直接用它的误差值来进行PID控制，生产主动轮电机的转速指令。但是通过仿真以后发现传感器读数与主动轮的转速（转角的差分）并不是线性的，意味着我们辛辛苦苦调出来的PID参数只是局部最优，当球在位于其他位置的时候PID算法的控制效果会大打折扣。因而我们有必要对传感器读数与主动轮转速的非线性关系补偿掉。现在我要做的就是把传感器读数与主动轮转角的关系找出来，然后对主动轮转角进行PID控制（转速的积分就是转角，直接的线性关系），可以叫“反馈线性化”，把非线性的反馈给线性化。

根据仿真得出的传感器读数-主动轮转角曲线，利用Matlab可以拟合出两者的函数关系为：

|  |
| --- |
| Linear model Poly4:  f(x) = p1\*x^4 + p2\*x^3 + p3\*x^2 + p4\*x + p5  Coefficients (with 95% confidence bounds):  p1 = 42.54 (41.28, 43.81)  p2 = -36.65 (-39.19, -34.1)  p3 = 12.16 (10.5, 13.81)  p4 = 33.38 (33, 33.76)  p5 = 0.06826 (0.04634, 0.09017) |

其中x为0~1的传感器读数，输出f(x)为0~51度的滚轮转角。传感器读数为0表示球完全脱离，滚轮转角用于生成线性的PID电机指令。

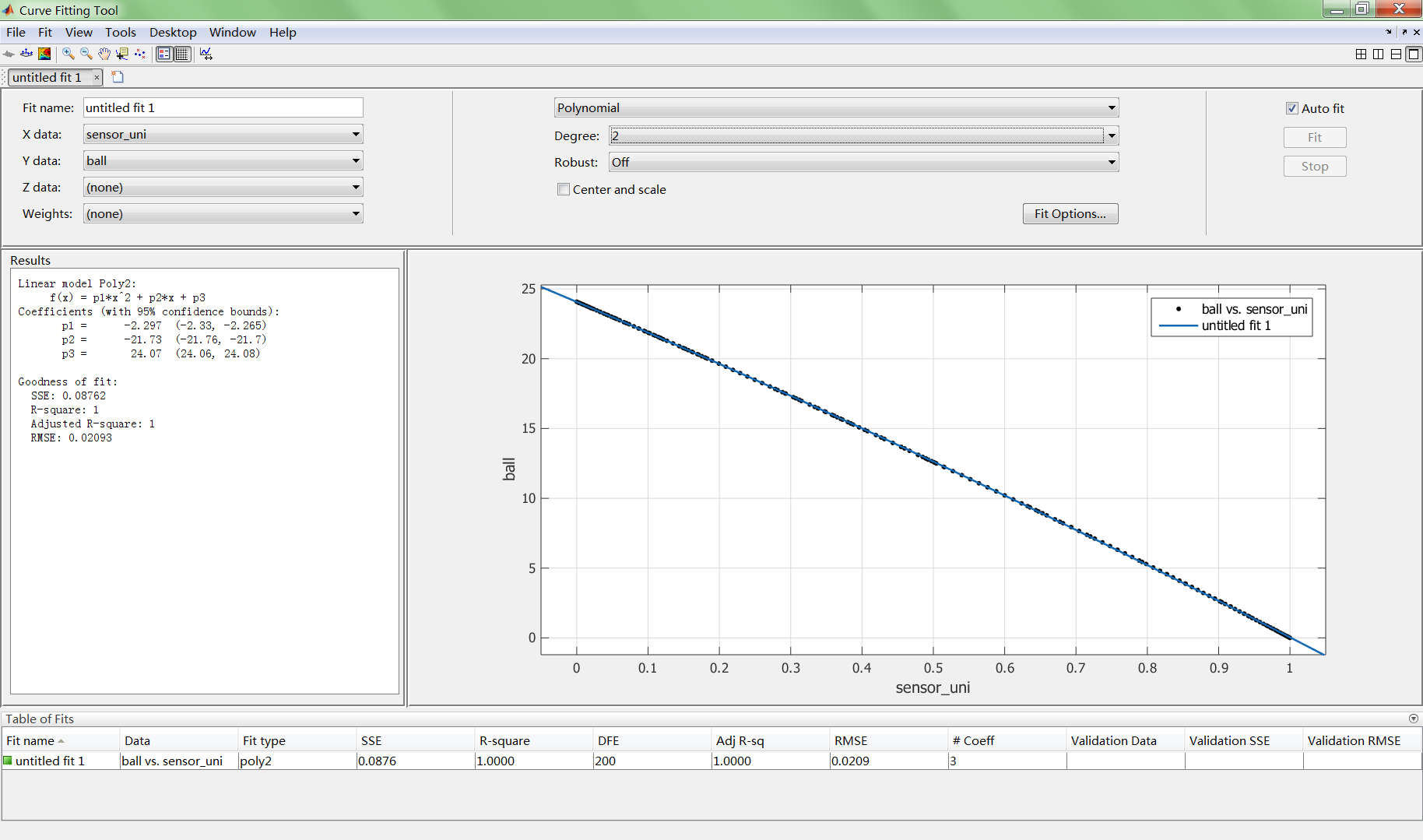


1. 直线位移传感器读数与球距

由于上层控制节点有时需要知道球的实际距离是多少，而机器得到的反馈是传感器的读数，因而有必要给出传感器读数与球距的转换关系。利用Matlab拟合出两者的函数为：

|  |
| --- |
| Linear model Poly2:  f(x) = p1\*x^2 + p2\*x + p3  Coefficients (with 95% confidence bounds):  p1 = -2.297 (-2.33, -2.265)  p2 = -21.73 (-21.76, -21.7)  p3 = 24.07 (24.06, 24.08) |

其中x为0~1的传感器读数，输出f(x)为24~0mm的球距。当球距小于24mm时，表示球还在主动带球的控制范围内，当它大于24mm时则理论上已经脱离了主动带球机构。



1. 球距与主动轮转角

因为上面（一）说了PID的参考量是主动轮转角，而我们的直观概念是球距，因而在设置PID目标点的时候需要知道球距与主动轮转角的转换关系。同样，利用Matlab拟合出的 球距 与 主动轮转角 的函数为：

|  |
| --- |
| Linear model Poly4:  f(x) = p1\*x^4 + p2\*x^3 + p3\*x^2 + p4\*x + p5  Coefficients (with 95% confidence bounds):  p1 = 7.302e-05 (7.125e-05, 7.479e-05)  p2 = -0.006544 (-0.00663, -0.006457)  p3 = 0.2102 (0.2088, 0.2115)  p4 = -4.426 (-4.434, -4.418)  p5 = 51.56 (51.54, 51.57) |

其中x为24~0mm的球距，输出f(x)为0~51度的滚轮转角。

