Model-Free Active Balancing for Humanoid Robots

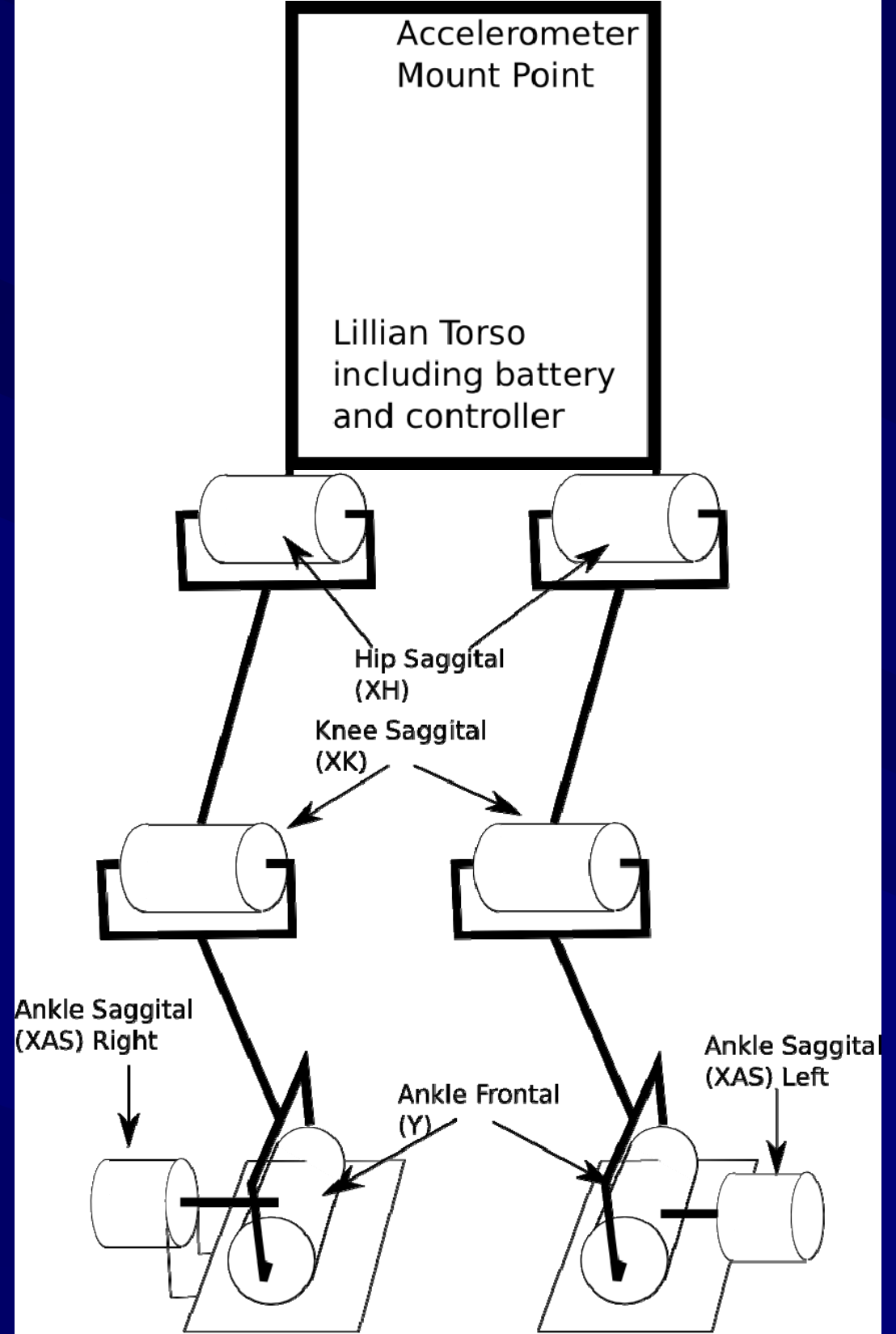
Sara McGrath, Jacky Baltes, and John Anderson

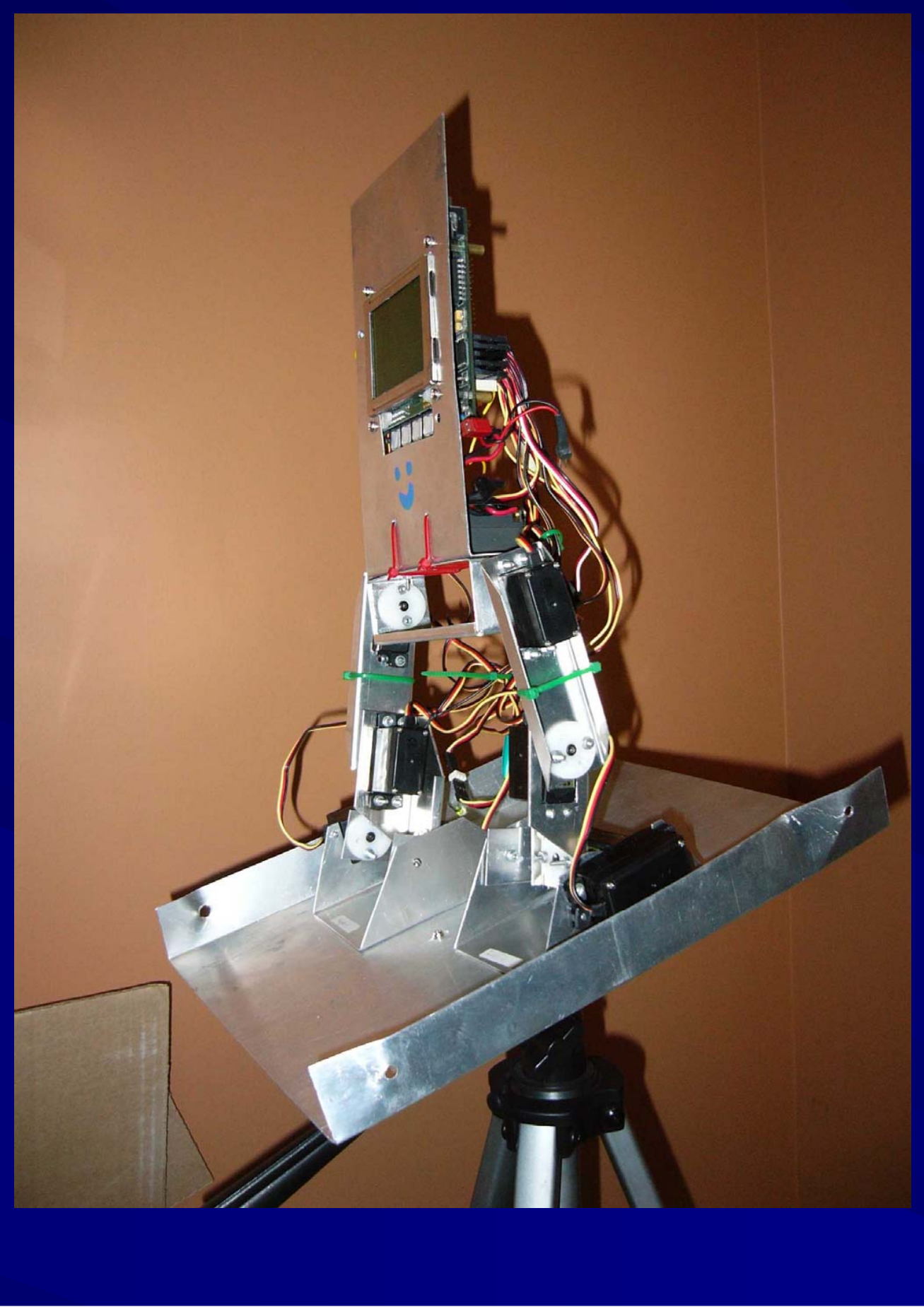
Autonomous Agents Laboratory, Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada R3T2N2

Introduction

- Practical humanoid robots require active balancing to move over a variety of terrain at variable speeds
- Adding active balancing will also allow robots to deal with sudden changes in equilibrium, and adaptation to new gaits and tasks (e.g. crawling, load-bearing) with greater ease and robustness
- Previous work (mostly in simulation and requiring an accurate mathematical model of the robot) uses combinations of sensors for feedback
- Little has been done on simple balancing reflexes employing a simple algorithm
- This work involved developing balancing reflexes using sensory types individually, to determine the extent of the utility of each

Humanoid Platform



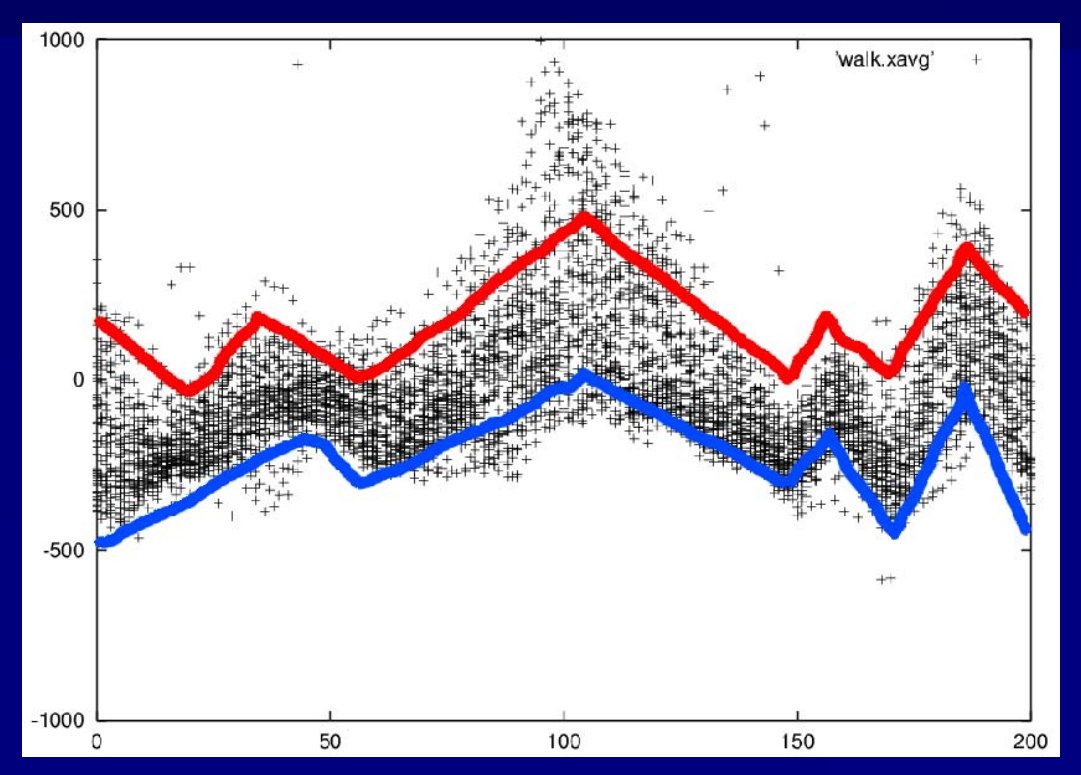


- Lillian, an 8DOF robot (4 DOF in each leg)
- The ankle moves in the sagittal plain (XAS joints) and the frontal plane (Y joints). One DOF in the knee (XK) and one in the hip (XH)
- Driven using an Eyebot controller, equipped with a Memsic 2125 2-axis, +/-1.5g accelerometer
- Main design principle is frugality: requires more robust algorithms and leads to more versatile solutions

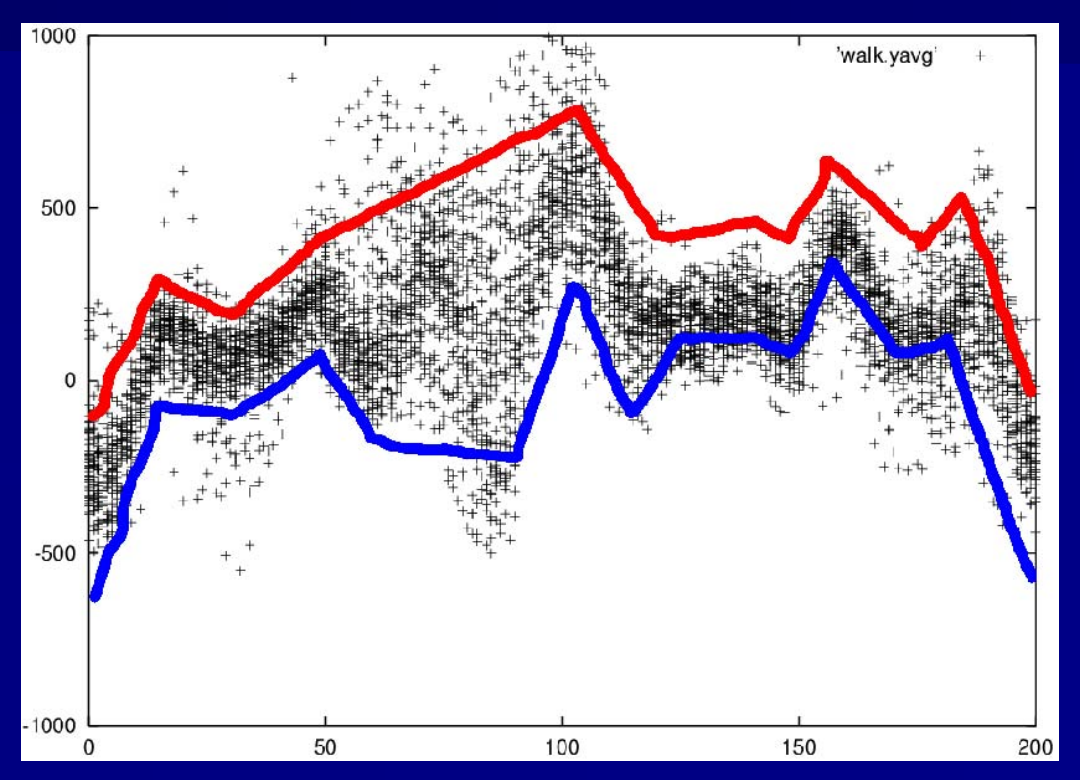
Baseline Walk

- Most approaches to humanoid balancing are model-based: a model of the robot is created and a control algorithm is implemented and tested on the model
- Our approach is implementation-based and does away with the need for a model: the robot is its own test platform
- Our methodology is to modify a pre-existing gait to improve it. This also allows the results of each strategy to be compared to one another and to the uncorrected walk
- This requires being able to record baseline motions (successful motions and their sensor readings) and perform common-sense analysis of the joint motions of the robot. E.g., X and Y plane s accelerometer readings for several dozen stops of a successful walk:

X



Y



- Most readings fall in a small band. Analyzing this walk produced upper (red) and lower (blue) threshold limits to be used in a threshold controller

Methodology

- Used 3 popular control algorithms to convert sensory readings into motion corrections to implement balancing: a PID controller, a threshold controller, and a hybrid of these
- For accelerometer-based PID control, a baseline is created by taking a sample from previously-programmed good motions, or setting the baseline to be unmoving (stand)
- Threshold balancing determines a threshold area where no corrections are made. This is centered on the PID baseline, but with a broader range. Corrections are only applied once an error value has been exceeded
- A hybrid controller was chosen for comparison in order to alleviate weaknesses with each of these. Our hybrid approach uses Thresholds for smaller corrections, but PID-based corrections for larger errors
- Sum of Absolute Error (SAE) is used to compare goodness of varying walks. All correction methods have a baseline/thresholds, so deviation from these can be measured to directly compare trials

Tuning

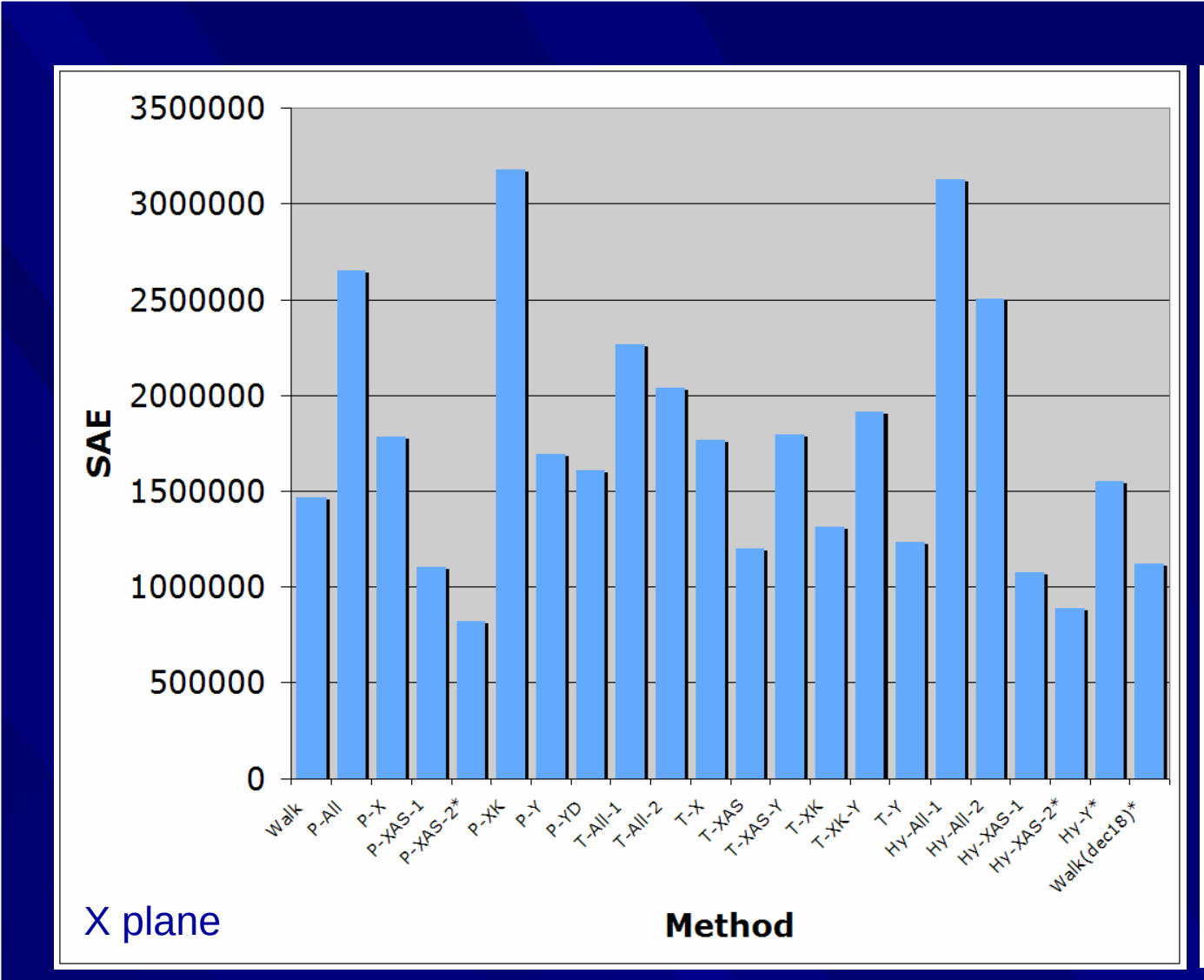
- Lillian was tuned in one plane before considering additional planes, and small increments of complexity were used
- First, corrective methods are tuned not to oscillate during stands, giving a minimal value to use as a base for PID settings/threshold bounds
- Then, the robot is tuned to stand still on a surface that tilts in either the frontal or sagittal plane, forcing corrections. Once the robot is stable while tilting, the robot is tuned using a a walk baseline on a flat surface

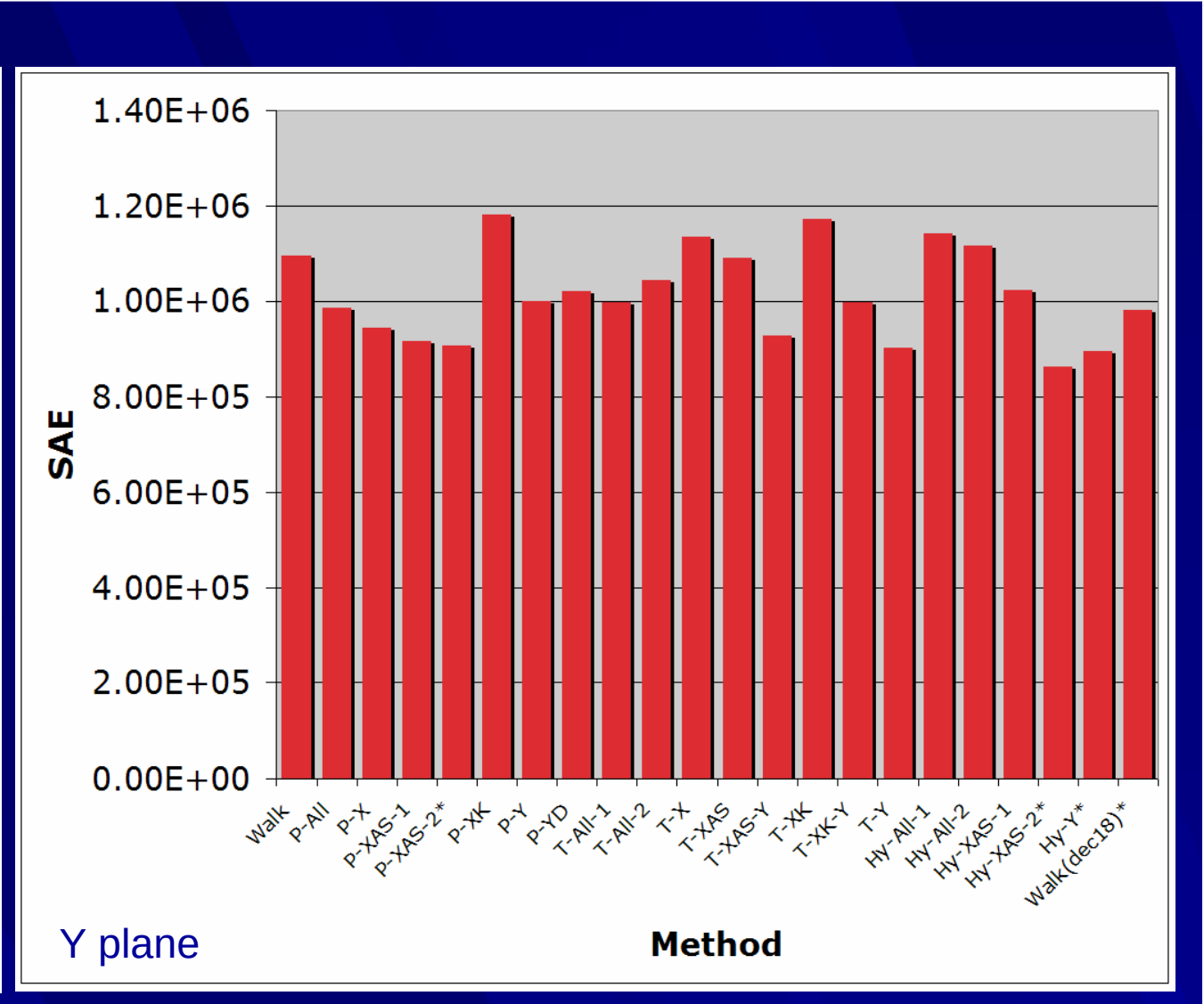
- Two axes are not tested until one is complete. Two-axis balancing tuning follows the same sequence as a single axis. Tuning began with PID and threshold methods and the best results from these were used for hybrid tuning
- Tuning on the tilting platform began with single joints, with platform angles from -30..+30, starting from 0, with an angular velocity of 240 per minute. Tests were coarse-grained, using controller gains of 150, 450, 750, and 1050, with delays of 10, 40, and 70ms between corrections
- Results from single joints were used to tune multiple joints

Evaluation

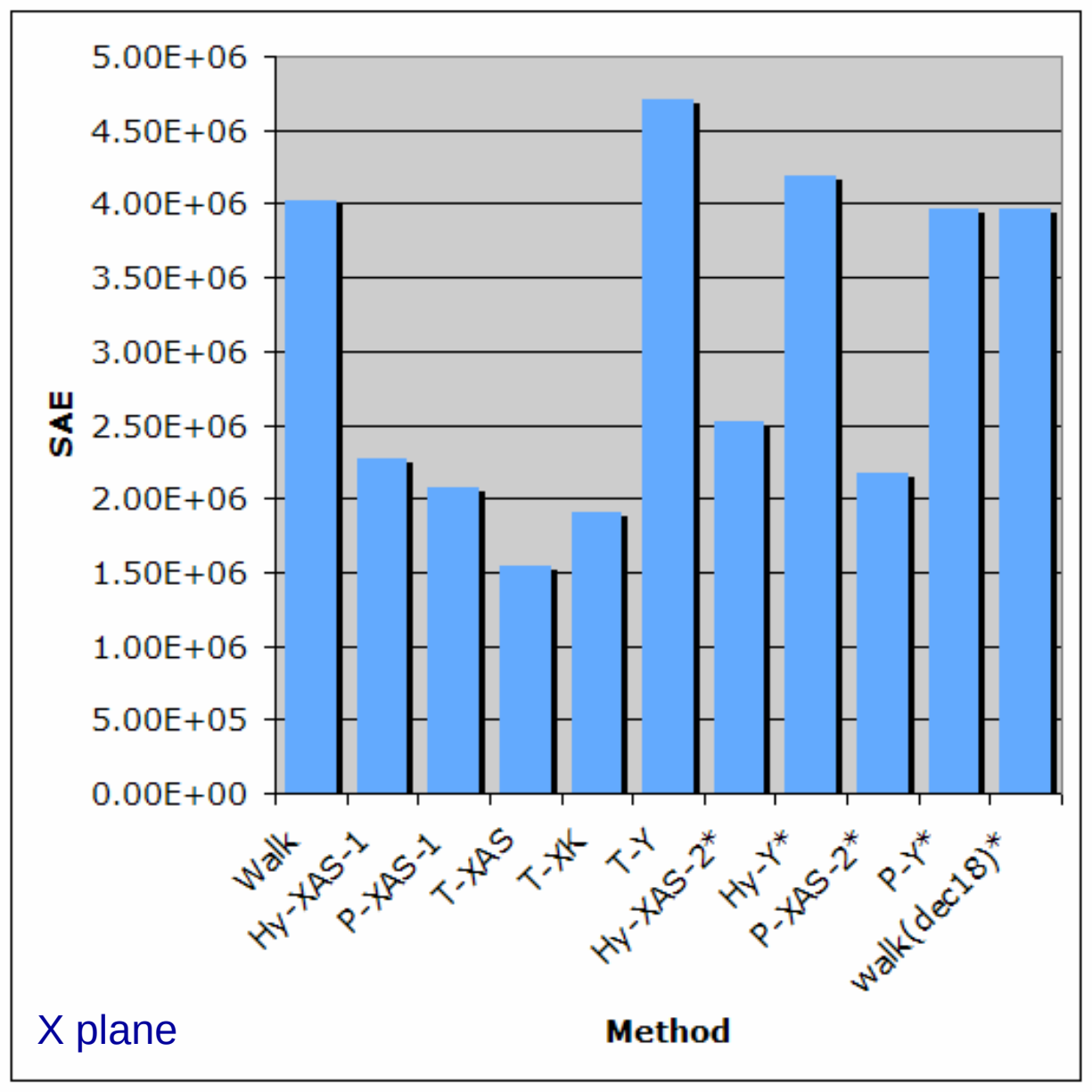
- A basic walking gait was used to evaluate the best tuning results side-by-side. Any result that improved on the gait was used for the final evaluation
- Gaits were perturbed by randomly varying the control points of one good gait over a spread of 5 or 10 set points ([-2...2] or [-5...5] at multiple points throughout the gait. Disturbances were applied to both joints
- PID controller settings chosen were the best for each joint, the best combination for X joints, the best for all joints, and the best P and PD controllers for Y. Threshold settings were similarly chosen for each joint and best combinations of X and XY joints. Other combinations were also chosen based in improvement of a basic walk
- The Hybrid controller was tested after the others, since it depends on combining these. This was tested on settings that improve the walk

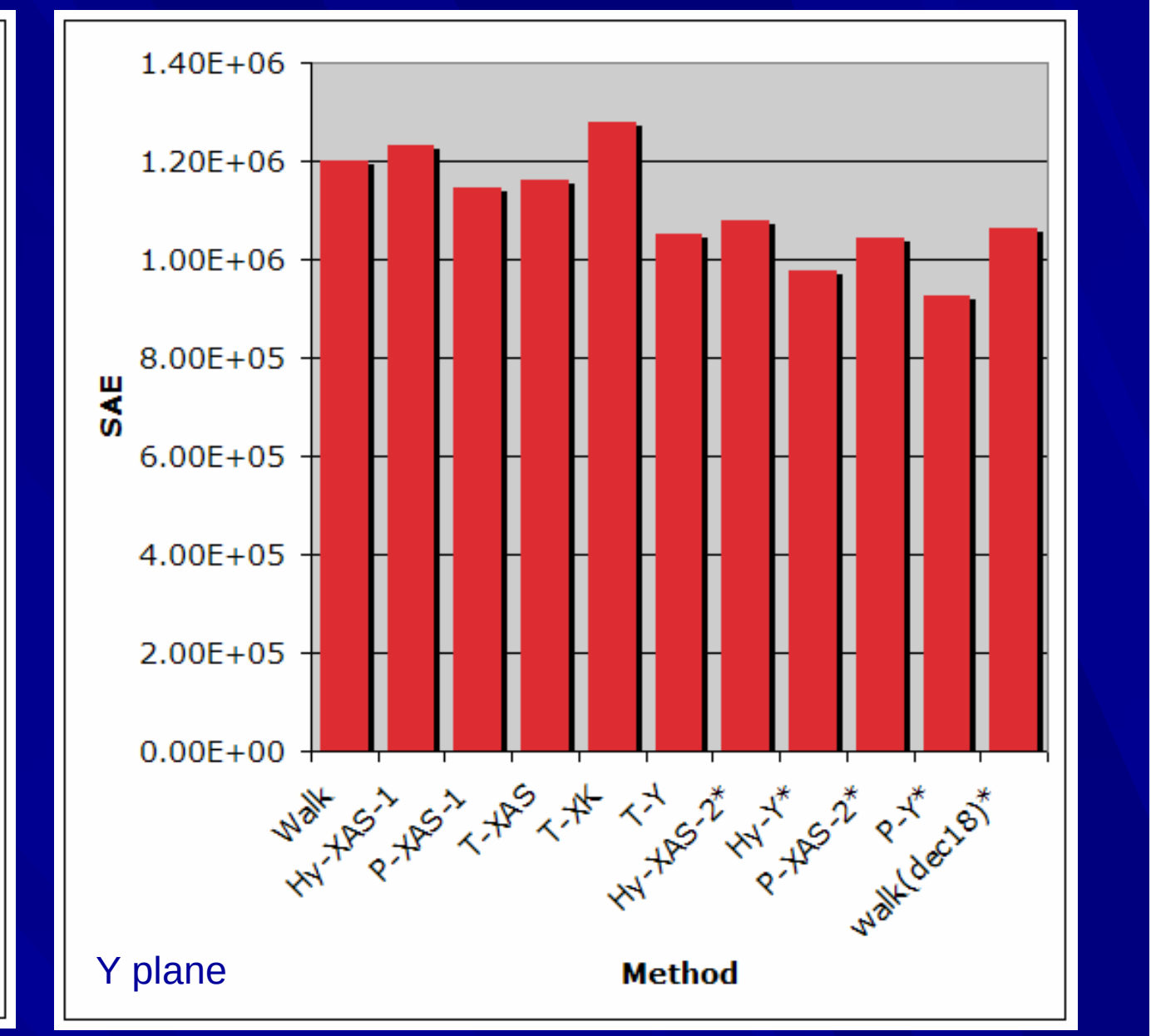
Results



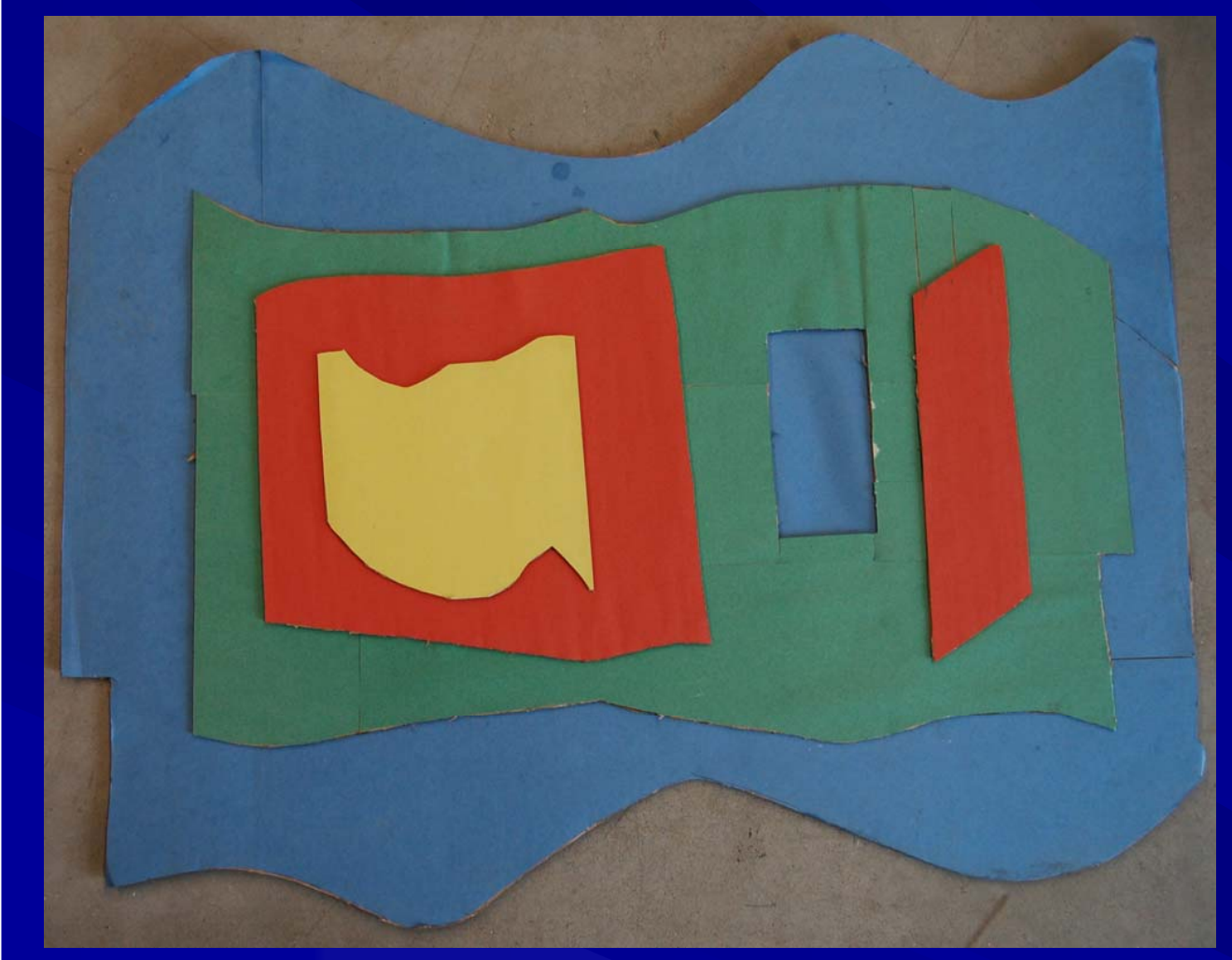


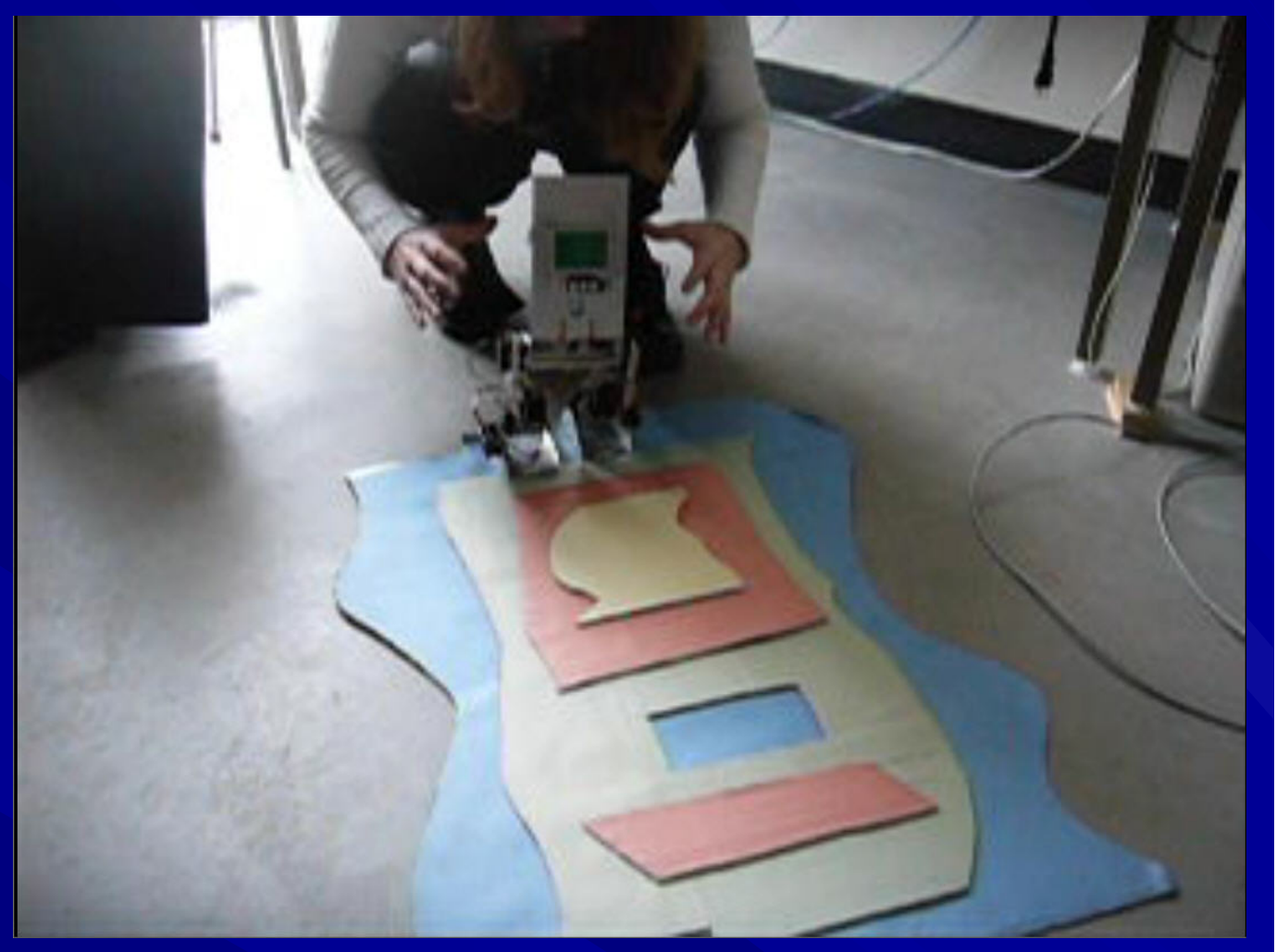
- Less than half of these factors (above) actually improve on the walk in the plane(s) they are correcting for. This was used to select settings for final evaluations: P-XAS-1, P-XAS-2, Hy-XAS-1, Hy-XAS-2, T-XAS, T-XK, T-Y, P-Y, and Hy-Y
- We then tested the algorithm by perturbing the walk (introducing error in the servo motors) by 5%, 10%, and 15%. While the need for correction was apparent even in a 5% case, the 10% case shows a greater difference between correction methods:



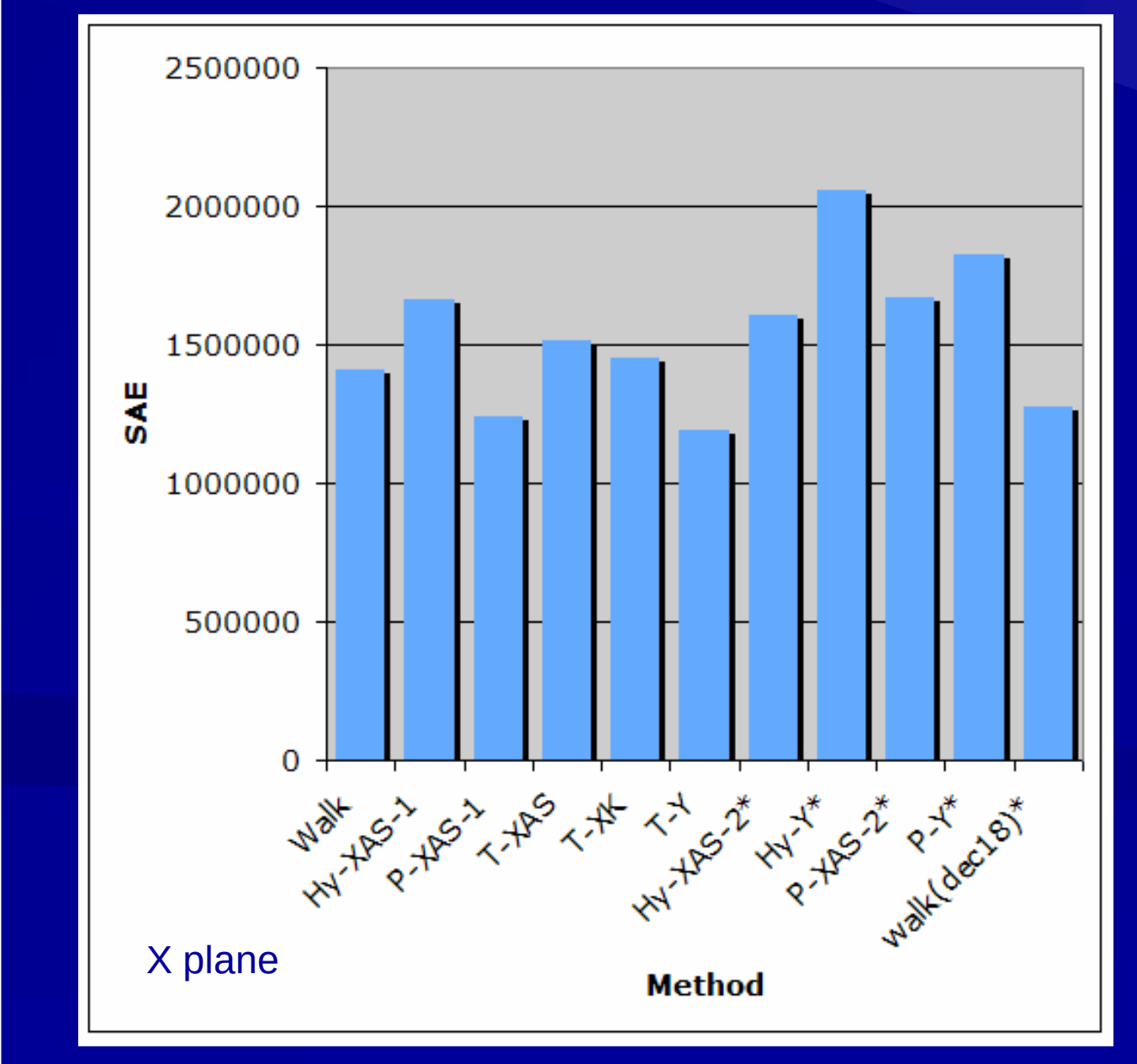


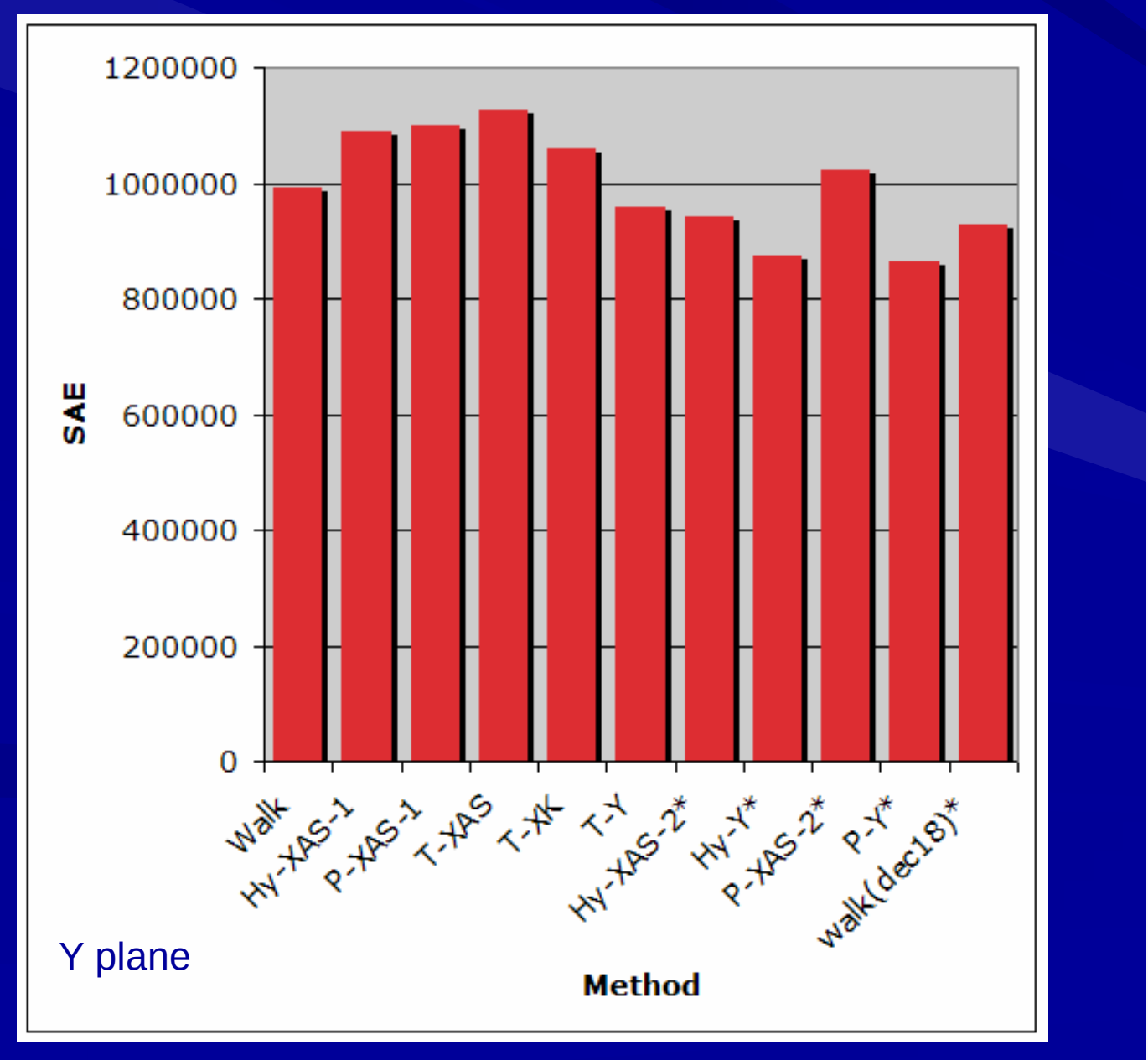
- Thresholds are best in the X plane, but with a greater difference between them than the other methods; All correction methods still show a marked improvement over uncorrected walk
- Finally, we used a stepping field (layered cardboard with a height difference of 3mm between pieces, but with >1 piece over the distance of a robot foot) to show that balancing degrades in more challenging situations.





- For the X plane, only the PID corrections improve on the uncorrected walk. Thresholds leave the walk mainly unchanged, while the hybrid method worsens the SAE readings





- Threshold and PID algorithms showed impressive results in tilting tests, but moving these to a walk demonstrated the shortcomings of reflex algorithms. No one algorithm was consistently best, but because of ease of tuning and implementation, threshold method is the most useful choice for future balancing work
- This work has focused on walking, but will be expanded in future to crawling and load-bearing