GRÖBNER BASES FOR POLYNOMIAL SYSTEMS IN ROBOTICS

Mina Khan[1], Frank Permenter[2] and Russ Tedrake[2]

[1]Mount Holyoke College, [2]Electrical Engineering and Computer Science, Massachusetts Institute of Technology

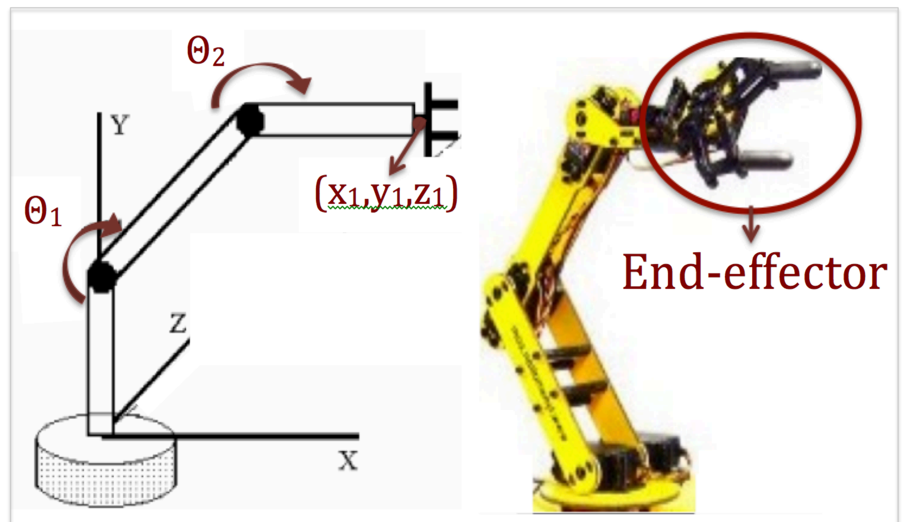
MSRP MIT Summer Research Program

CSAIL

Motivation: *Efficiently* find *exact* solutions to multivariate polynomial equations in robot control and motion planning.

I. ROBOT CONTROL

- Robot control involves solving polynomial equations in inverse kinematics (positions) and dynamics (forces) problems for robots.



Serial N-link manipulator model (left) for a robot arm (right)

II. GRÖBNER BASES

✔ Gröbner bases can be used to find solutions to systems of multivariate polynomial equations.

✖ Gröbner basis algorithms have double exponential worst-case complexity, so we aim to find efficient Gröbner basis algorithms.

Polynomial System

$f_1(x1, x2, x3,....,xn) = 0$
$f_2(x1, x2, x3,....,xn) = 0$
$f_2(x1, x2, x3,....,xn) = 0$
:
$f_m(x1, x2, x3,....,xn) = 0$

Gröbner basis

$g_1(x1) = 0$
$g_2(x1, x2) = 0$
$g_3(x1, x2, x3) = 0$
:
$g_m(x1, x2, x3,....,xn) = 0$

INPUT → Gröbner basis algorithm → OUTPUT

When a system has a unique solution, Gröbner basis has at least one univariate equation and subsequent equations with an increasing number of variables[1]. Thus, we can solve Gröbner basis easily by back-substitution.

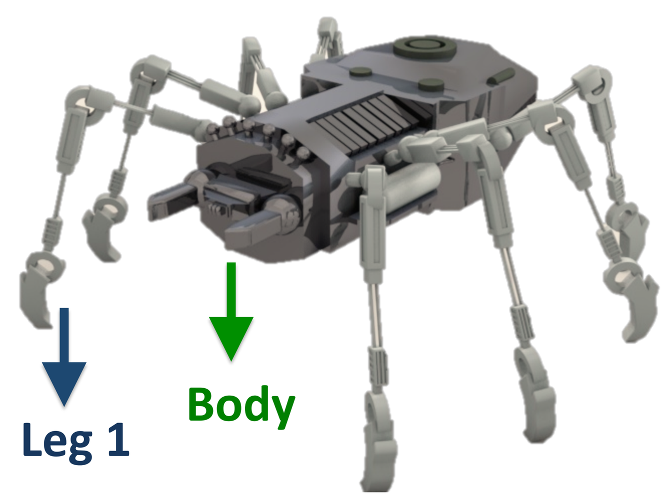
III. ROBOT STRUCTURES

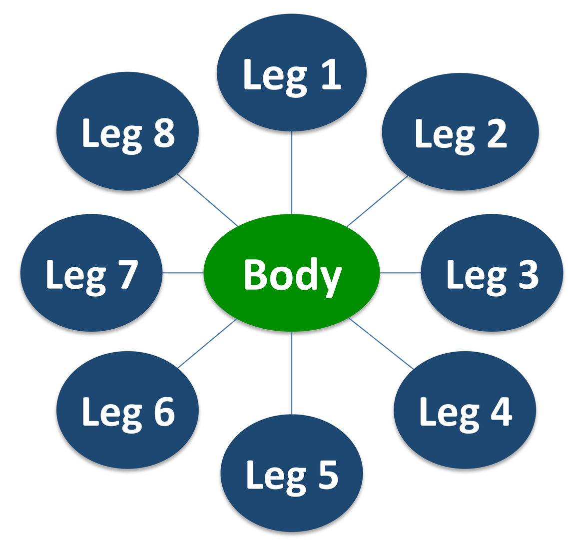
- Sparse matrix computations are simpler than dense matrix computations.

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{vmatrix} = 24$$

Determinant of a sparse diagonal matrix

We created a spider robot (left). Each spider leg is connected to the body, but not to other legs, yielding chordal sparsity in robot graphs (right).





IV. COMPUTATIONS

- Gröbner basis algorithms include Buchberger's algorithm, Faugère's F4, F5 and FGLM.

✖ Maple, MATLAB and Macaulay2 *cannot* compute Gröbner basis for a polynomial system representing even one robotic spider leg.

Maple 18 MATLAB Macaulay2

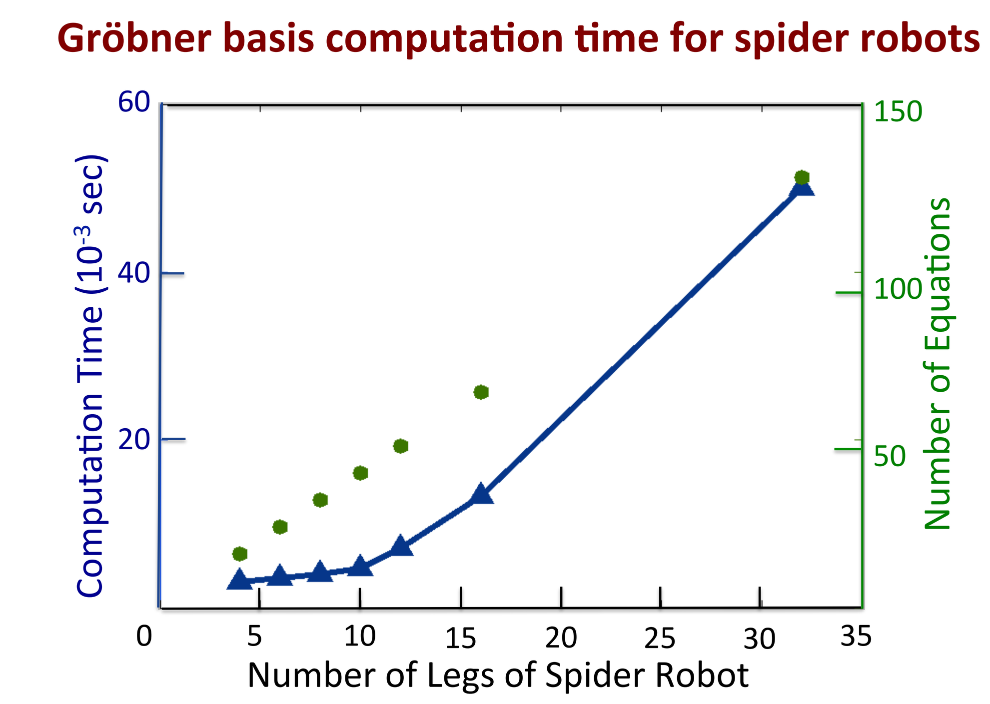
- Sage is Python-based math software that uses C++ libraries in Singular and Magma to compute Gröbner bases.

✔ Sage allows fast Gröbner basis computations.

sage python MAGMA COMPUTER • ALGEBRA C C++ SINGULAR

V. CONCLUSIONS

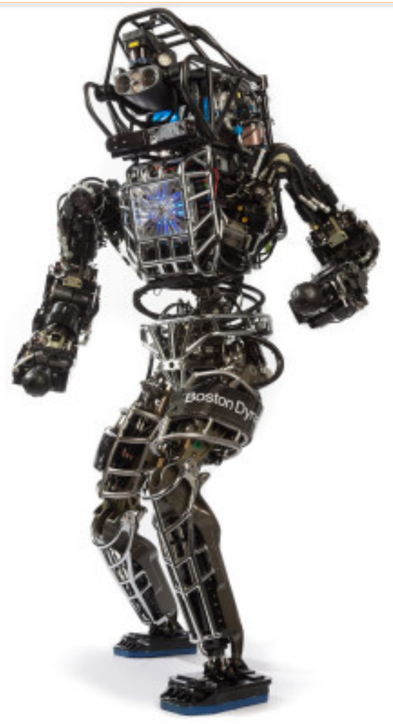
- Sage efficiently computes Gröbner bases using a 1.3 GHz Intel Core i5 processor. We achieved a 50 milliseconds computation time for Gröbner basis for a system of 128 polynomials in 67 variables for a 32-leg spider robot.



Computation time for Gröbner bases using Sage. Our polynomial systems for the spider robot had 16 to 128 equations.

VI. FUTURE WORK

- Integrate Sage's Gröbner basis algorithms in our existing polynomial solvers to improve the performance of robots, especially Boston Dynamics "Atlas".
- Investigate chordal graphs to further leverage polynomial sparsity in Gröbner basis algorithms. Our existing algorithm using chordal graphs is slower than Sage's algorithms.



Atlas robot

Acknowledgements

Robot Locomotion Group, Diego Cifuentes, and MSRP Staff

Citation: [1] arXiv:1108.1201

Picture Courtesy: ROSARIO, J. M. et al. Proposal of methodology for the modeling and control of manipulators. J. Braz. Soc. Mech. Sci. 2002, vol.24, n.3; http://www.societyofrobots.com/robot_arm_tutorial.shtml; http://physicssum.deviantart.com/art/3D-Robot-Spider-280498884; Logos: MATLAB, Maple, Macaulay2, Sage, Python, Magma, Singular, C/C++ (http://www.techhui.com/group/candcppdevelopers); http://www.fairfaxunderground.com/forum/file.php?40,file=113550,filename=Atlas-p2_nt.jpg