



Inverse Kinematic Solution for Romin Robot Based on Computer Algebra Techniques

Mahdi Dehghani Darmian¹

¹Assistant Professor, Department of Basic Sciences, Technical and Vocational University (TVU), Tehran, Iran.

ARTICLE INFO

Article Type:

Original Research

Received: 04.07.2024

Revised: 07.22.2024

Accepted: 09.14.2024

Keyword:

Inverse kinematics
Romin robot
Monomial ordering
Gröbner Bases
Gröbner system
Maple

***Corresponding Author:**

Mahdi Dehghani Darmian

Email:

mdehghanidarmian@tvu.ac.ir

m.dehghanidarmian@ipm.ir

ABSTRACT

This paper employs computer algebra techniques to solve the inverse kinematics problem of the 3D Romin Robot. Similar to the approach applied to simple planar robots, which is based on analyzing the parameter space without considering any details, we extend it to the 3D Romin Robot by encompassing all conceivable parameters related to the angles and arm lengths of this robot. In particular, we apply the GES-GVW-CGS algorithm (an efficient algorithm for computing the Gröbner systems) to partition the parameter space into a finite set of parametric cells and assign a set of parametric polynomials to each cell. We solve the inverse kinematics problem according to each cell, which contains specific parameter information. All algorithms discussed in this paper are implemented in Maple software. Compared to traditional methods, the proposed approach significantly improves the solution to the inverse kinematics problem for the 3D Romin Robot. By leveraging computer algebra techniques and the GES-GVW-CGS algorithm within the Maple software environment, we can efficiently analyze the parameter space and provide solutions for a wide range of robot configurations. This research expands the applicability of existing methods to complex robotic systems and showcases the power of computational tools in advancing robotics research and development.

EXTENDED ABSTRACT

Introduction

The problem of inverse kinematics in a robot manipulator entails determining the joint angles for a specific position and orientation of the manipulator's hand. This problem can be formulated algebraically as solving a system of polynomial equations with parametric coefficients. The variables represent the joint angles, while the parameters correspond to the hand's location. Utilizing Gröbner systems for solving polynomial systems with parametric coefficients offers a practical approach to tackling this problem. By obtaining a finite set of triangular parametric polynomial systems, along with parametric conditions, we can simplify the original problem. It is important to mention that González-López et al. manually address this problem using the "dynamic evaluation method." In their approach, they combine algebraic techniques with geometric reasoning to efficiently determine the joint angles required for a given end-effector position and orientation. The Gröbner system method allows for a systematic and structured way of solving the inverse kinematics problem, providing a reliable solution for various manipulator configurations.

Methodology

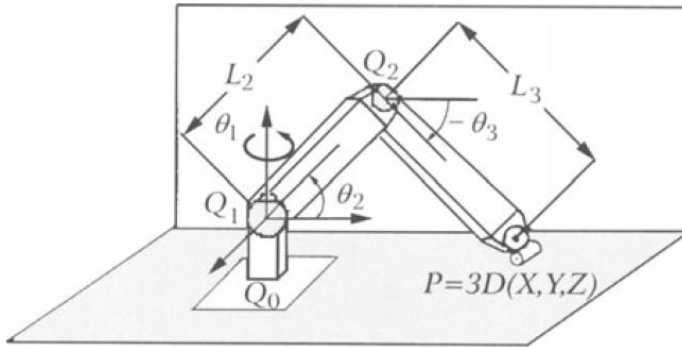
The Gröbner system, as a computer algebra method, can be used to solve the inverse kinematics problem for the ROMIN robot, which is employed in production lines and product packaging. Additionally, the Romin robot is utilized in pressurized water nuclear reactors for periodic inspections and repairing tubes inside the reactor. The inverse kinematics problem employs kinematic equations to determine the robot's motion to reach a desired position. As these equations rely on certain parameters and variables, employing parametric Gröbner bases proves useful in solving these parametric equations to identify potential routes to a fixed position. Gröbner bases are a specific type of generator set of a polynomial ideal and serve as a potent tool in computer algebra. Introduced in 1965 and named after his advisor, Wolfgang Gröbner. He proposed two criteria to enhance the performance of his algorithm. Lazard described an algorithm for computing Gröbner bases by using linear algebra techniques. Faugère contributed to the development of two renowned algorithms, F4 and F5, for computing Gröbner bases. Given our focus on parametric Gröbner bases, particularly the Gröbner system, we delve into the existing literature on this subject. The concept of Gröbner systems and the initial algorithm for computing them were introduced by Weispfenning in 1992. Over the years, significant progress and numerous results have been achieved in computing Gröbner systems, culminating in Dehghani's 2024 proposal of an efficient algorithm, known as the GES-GVW-CGS algorithm, for their computation.

Results and discussion

Gröbner systems find various applications in Mathematics and other scientific fields. Specifically, we highlight their role in solving the inverse kinematics problem of the Romin robot. The Romin robot with three arms (the length of the first arm is invariant) and three degrees of freedom is located in \mathbb{R}^3 . The angle θ_1 rotates around an axis perpendicular to

the ground. Also, θ_2 (resp., θ_3) is the angle measured from the horizontal plane Q_1 (resp., Q_2) to the second arm (resp., third arm). For any point $(x, y, z) \in \mathbb{R}^3$ and also by considering $d^2 = x^2 + y^2$ and utilizing the 3 matrix rotation, we obtain the following subset, which can be considered as the algebraical model describing the geometry of the Romin robot:

$$F = \{c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1, c_3^2 + s_3^2 - 1, x + ds_1, y - dc_1, l_2c_2 + l_3c_3 - d, l_2s_2 + l_3s_3 - z\}.$$



By utilizing the Maple implementation of the GES-GVW-CGS algorithm, a Gröbner system for F is computed, featuring 34 branches, with 22 being generic and 12 being singular. In cases of singularity, the Romin robot lacks solutions, showcasing the success of the GES-GVW-CGS algorithm in automatically computing all singular instances. The details of the singular branches are presented in the following tables:

Null Sets	Not Null Sets	Gröbner Bases
$[d, z, x, l_3, l_2]$	$[y]$	$[1]$
$[d, x, l_3, l_2]$	$[y, z]$	$[1]$
$[d, x, l_3]$	$[l_2, y]$	$[1]$
$[d, x]$	$[l_3, y]$	$[1]$
$[d, z, y, l_3, l_2]$	$[x]$	$[1]$
$[d, y, l_3, l_2]$	$[x, z]$	$[1]$
$[d, y, l_3]$	$[l_2, x]$	$[1]$
$[d, y]$	$[l_3, x]$	$[1]$
$[d, z, l_3, l_2]$	$[x, y]$	$[1]$
$[d, l_3, l_2]$	$[x, y, z]$	$[1]$
$[d, l_3]$	$[l_2, x, y]$	$[1]$
$[d]$	$[l_3, x, y]$	$[1]$

For instance, let us consider one of the generic branches of the computed Gröbner system as shown below:

$$N = [x^2 + y^2 - d^2]$$

$$W = [d, l_2, l_3, d^2 + z^2]$$

$$G = \left[\begin{array}{l} ds_1 + x, c_1d - y, c_2^2 + s_2^2 - 1, l_2s_2 + l_3s_3 - z, \\ c_2l_2 + l_3 - d, -2c_2dl_2 - 2l_2s_2z + d^2 + l_2^2 - l_3^2 + z^2, \\ 4d^2l_2^2s_2^2 + 4l_2^2s_2^2z^2 - 4d_2l_2s_2z - 4l_2^3s_2z + 4l_2l_3^2s_2z - 4l_2s_2z^3 + \\ d^4 - 2d^2l_2^2 - 2d^2l_3^2 + 2d^2z^2 + l_2^4 - 2l_2^2l_3^2 + 2l_2^2z^2 + l_3^4 - 2l_3^2z^2 + z^4 \end{array} \right]$$

Let's analyze and examine this triple where $N = [x^2 + y^2 - d^2]$ and $W = [d, l_2, l_3, d^2 + z^2]$ represent null and non-null sets, respectively. This implies that if $x^2 + y^2 - d^2 = 0$ and $d \neq 0, l_2 \neq 0, l_3 \neq 0, d^2 + z^2 \neq 0$, the set G below is a Gröbner basis with respect to the parametric constraints:

$$G = \left[\begin{array}{l} ds_1 + x, c_1d - y, c_2^2 + s_2^2 - 1, l_2s_2 + l_3s_3 - z, \\ c_2l_2 + l_3 - d, -2c_2dl_2 - 2l_2s_2z + d^2 + l_2^2 - l_3^2 + z^2, \\ 4d^2l_2^2s_2^2 + 4l_2^2s_2^2z^2 - 4d_2l_2s_2z - 4l_2^3s_2z + 4l_2l_3^2s_2z - 4l_2s_2z^3 + \\ d^4 - 2d^2l_2^2 - 2d^2l_3^2 + 2d^2z^2 + l_2^4 - 2l_2^2l_3^2 + 2l_2^2z^2 + l_3^4 - 2l_3^2z^2 + z^4 \end{array} \right]$$

After replacing the parameter values $l_2 = 3.5, l_3 = 5.5, (x, y, z) = (3, 4, 6) \in \mathbb{R}^3$ in the Gröbner basis G that satisfies N and W, we obtain the following set of polynomials:

$$\left\{ \begin{array}{l} 5s_1 + 3 = 0 \\ 5c_1 - 4 = 0 \\ c_2^2 + s_2^2 - 1 = 0 \\ 5.5s_3 + 3.5s_2 - 6 = 0 \\ 5.5c_3 + 3.5c_2 - 5 = 0 \\ -35c_2 - 42s_2 + 43 = 0 \\ 2989s_2^2 - 361s_2 + 624 = 0 \end{array} \right.$$

Upon solving the system comprising 7 equations, the outcomes are as follows:

$$\left\{ \begin{array}{l} s_3 = 0.9580021314 \quad \text{and} \quad c_3 = 0.2867610787 \\ s_2 = 0.2088537935 \quad \text{and} \quad c_2 = 0.9779468763 \\ s_1 = -0.6 \quad \text{and} \quad c_1 = 0.8 \end{array} \right.$$

$$\left\{ \begin{array}{l} s_3 = 0.4548145601 \quad \text{and} \quad c_3 = 0.8905861642 \\ s_2 = 0.9995771198 \quad \text{and} \quad c_2 = 0.02907888479 \\ s_1 = -0.6 \quad \text{and} \quad c_1 = 0.8 \end{array} \right.$$

Consequently, corresponding to these solution sets, two distinct scenarios emerge for the angle between the arms:

$$\left[\begin{array}{l} \theta_3 = 73.333585444^\circ \quad \mathbf{and} \quad \theta_2 = 12.05519009^\circ \quad \mathbf{and} \quad \theta_1 = -36.86989764^\circ \\ \theta_3 = 27.053003323^\circ \quad \mathbf{and} \quad \theta_2 = 88.33366771^\circ \quad \mathbf{and} \quad \theta_1 = -36.86989764^\circ \end{array} \right.$$

Conclusion

Solving the problem of inverse kinematics in robots is similar to many other interesting applications that can be explored using the Gröbner system. These systems can be utilized in image processing, cryptography, biochemistry, automated geometry proving, and other fields where complex algebraic problems necessitate the analysis of different conditions. Through the use of computer algebra techniques, researchers and engineers can effectively analyze intricate calculations and discover solutions to practical issues. The versatility and analytical capabilities of Gröbner systems have established them as valuable tools in various scientific and engineering domains. An exemplary application of computer algebra techniques in engineering sciences is the automatic calculation and analysis of all singularities in the studied Romin robot.



کارافن

فصلنامه علمی دانشگاه ملی مهارت

ویژه‌نامه ۱۴۰۴، دوره ۲۲، ۵۸۰-۶۰۶

آدرس نشریه: <https://karafan.tvu.ac.ir/>

doi: [10.48301/kssa.2024.450007.2876](https://doi.org/10.48301/kssa.2024.450007.2876)



حل مساله سینماتیک معکوس ربات رومین بر پایه تکنیک‌های جبر محاسباتی

مهدی دهقانی درمیان^۱

۱- استادیار، گروه علوم پایه، دانشگاه ملی مهارت، تهران ایران.

اطلاعات مقاله	چکیده
نوع مقاله: مقاله پژوهشی (کاربردی)	این مقاله از تکنیک‌های جبر کامپیوتری برای حل مسئله سینماتیک معکوس ربات رومین سه بعدی استفاده می‌کند. این روش مشابه روشی است که برای ربات‌های مسطح ساده اعمال می‌شود که بدون تجزیه و تحلیل فضای پارامتر و عاری از هرگونه جزئیات است. با این تفاوت که ما روش ربات ساده را با در نظر گرفتن تمام حالات پارامترهای قابل تصور مربوط به زوایا و طول بازوهای این ربات، به ربات رومین سه‌بعدی تعمیم می‌دهیم. به طور خاص، ما از الگوریتم GES-GVW-CGS که یک الگوریتم کارآمد برای محاسبه دستگاه‌های گریز است برای تقسیم فضای پارامتری به مجموعه محدودی از سلول‌های پارامتری استفاده می‌کنیم و مجموعه‌ای از چندجمله‌ای‌های پارامتری را به هر سلول اختصاص می‌دهیم و با توجه به هر سلولی که حاوی اطلاعات پارامتری خاصی است، مسئله سینماتیک معکوس را حل می‌کنیم. تمام الگوریتم‌های مورد بحث در این مقاله را در Maple پیاده‌سازی کرده‌ایم. رویکرد پیشنهادی پیشرفت قابل توجهی را در حل مسئله سینماتیک معکوس برای ربات رومین سه بعدی در مقایسه با روش‌های سنتی نشان می‌دهد. با استفاده از تکنیک‌های جبر کامپیوتری و الگوریتم GES-GVW-CGS در محیط نرم‌افزار Maple، می‌توانیم فضای پارامترها را به‌طور کارآمد تجزیه و تحلیل کنیم و راه‌حلی برای طیف وسیعی از پیکربندی‌های ربات ارائه کنیم. این تحقیق نه تنها کاربرد روش‌های موجود را در سیستم‌های رباتیک پیچیده گسترش می‌دهد، بلکه قدرت ابزارهای محاسباتی را در پیشرفت تحقیق و توسعه رباتیک به نمایش می‌گذارد.
کلید واژگان: سینماتیک معکوس ربات رومین ترتیب تک جمله‌ای پایه‌های گرنبر دستگاه گرنبر میپل	
نویسنده مسئول: مهدی دهقانی درمیان پست الکترونیکی: mdehghanidarmian@tvu.ac.ir m.dehghanidarmian@ipm.ir	



©2024 the authors. Published by National University of Skills, Tehran, Iran. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution-Noncommercial 4.0 International (CC BY-NC License) (<https://creativecommons.org/licenses/by-nc/4.0/>)

شاپای الکترونیکی: ۲۵۳۸-۴۴۳۰
شاپای چاپی: ۲۳۸۲-۹۷۹۶

مقدمه:

مسئله سینماتیک معکوس^۱ در رباتیک یک فرآیند ریاضی متشکل از معادلات و روابط سینماتیکی است که در تعیین حرکت ربات، محاسبه موقعیت‌های مفصلی و تعیین طول بازوها برای رسیدن به موقعیت مطلوب و مورد نظر در صفحه یا فضا مطرح می‌شود. برخلاف سینماتیک مستقیم^۲ که اگر مفاصل در یک موقعیت خاص تنظیم شوند، نقطه دسترسی ربات یک پاسخ خواهد داشت، سینماتیک معکوس، اغلب چندین راه حل مختلف و رویکرد متعدد برای دسترسی به نقطه مشخص ارائه می‌کند. هدف از حل مساله سینماتیک معکوس تعیین طول بازوها و مقادیر زاویه‌های اتصالات و مفاصل ربات برای قرار دادن دست ربات در یک نقطه و جهت معین از فضای دسترسی ربات می‌باشد. در برخی از ربات‌ها مساله سینماتیک معکوس با روش‌های تحلیلی و ریاضی (با محوریت ماتریس ژاکوبی^۳) قابل حل می‌باشد. در این مقاله قصد داریم مساله سینماتیک معکوس ربات رومین^۴ را با کمک مفاهیم و تکنیک‌های جبر کامپیوتری حل نماییم. شایان ذکر است این مساله با کمک روش ارزیابی پویا^۵ در [۱] مورد مطالعه قرار گرفته است و حالت‌های مفاصل و طول بازوها به صورت تک تک و غیر اتومات بررسی شده‌اند. ربات رومین یک ربات با درجه آزادی سه چرخشی است که دارای سه بدنه است که همگی دارای اتصالات چرخشی مسطح، پایه چرخشی و دو بازوی مفصلی با سه درجه آزادی هستند. این ربات همچنین دارای یک دست با دو درجه آزادی است. اکنون ساختار و ظاهر ربات رومین با توجه به توسعه پروژه‌ها و به دلایل فیزیکی تغییر کرده است. هدف ما بدست آوردن جواب مدل معکوس این ربات با استفاده از روش‌های دقیق نمادین و جبر کامپیوتری می‌باشد. ربات رومین دارای کاربردهای فراوانی است که مهمترین آن‌ها استفاده در خط تولید محصولات مختلف



شکل ۱: ربات بسته‌بندی خط تولید

^۱ Inverse kinematics problem

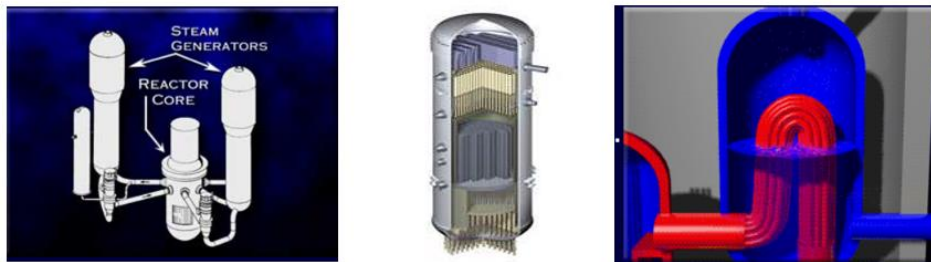
^۲ Forward kinematics

^۳ Jacobian matrix

^۴ Romin robot

^۵ Dynamic evaluation method

و بسته بندی آن‌ها و از همه مهمتر در مولدهای بخار در نیروگاه های هسته ای از نوع "راکتور هسته ای آب تحت فشار"^۱ است.



شکل ۲: مولدهای بخار در نیروگاه هسته‌ای

مولدهای بخار دارای دو جعبه آب مجزا هستند: یکی برای آب ورودی و دیگری برای آب خروجی که با تعداد بسیار زیادی لوله یو شکل (حدود ۵۰۰۰ لوله) به هم متصل شده‌اند و وظیفه آن‌ها گرم کردن مدار ثانویه راکتور می‌باشد. آب سرد بدون آلایندگی هسته‌ای برای بدست آوردن بخار آب در مدار اولیه از جعبه‌های آبی عبور می‌کند که حاوی محصولات رادیواکتیویته‌ای هستند و از راکتور هسته‌ای می‌آیند. پس این جعبه‌ها به شدت رادیو اکتیویته می‌شوند و لازم است که از تبادل آب بین مدارهای اولیه و ثانویه جلوگیری شود. مولد بخار چنین کار سختی را انجام می‌دهد (تحت فشار و ارتعاش زیاد) که با بررسی دوره‌ای برای حفظ وضعیت لوله‌های یو شکل انجام می‌شود. این عملیات باید از داخل جعبه‌های هسته‌ای آب انجام شود که با توجه به ابعاد کوچک آنها و داشتن رادیواکتیویته شدید این کار توسط انسان امکان‌پذیر نیست و توسط ربات رومین انجام می‌شود.

برای حل مساله سینماتیک معکوس در ابتدا باید یک مدل‌سازی جبری از ربات رومین حاصل شود. از آنجایی که معادلات حاصل از این مدل‌سازی به برخی پارامترها و متغیرها بستگی دارند، استفاده از پایه‌های گریبنر^۲ و پایه‌های گریبنر پارامتری برای حل این معادلات با هدف یافتن همه راه‌های ممکن برای رسیدن به آن موقعیت ثابت مفید است. همچنین با استفاده از پایه‌های گریبنر پارامتری می‌توانیم نقاط تکین^۳ را استخراج کنیم. مفهوم پایه‌های گریبنر نوع خاصی از مجموعه مولد یک ایده آل چند جمله‌ای است و ابزار قدرتمندی در جبر کامپیوتری است. پایه‌های گریبنر در سال ۱۹۶۵ همراه با یک الگوریتم (الگوریتم بوخبرگر) برای محاسبه آنها توسط برونو بوخبرگر^۴ در رساله دکترایش [۲] معرفی شد. او آنها را به افتخار استاد راهنمایش (ولفگانگ گریبنر^۵) پایه گریبنر نامگذاری کرد. سپس، او در [۳] دو معیار را برای بهبود عملکرد الگوریتم خود پیشنهاد کرد. در سال ۱۹۸۳، لازارد^۶ الگوریتمی را

^۱ Pressurized Water Reactor - PWR

^۲ Gröbner bases

^۳ Singular point

^۴ Bruno Buchberger

^۵ Wolfgang Gröbner

^۶ Lazard

برای محاسبه پایه های گرینر با استفاده از تکنیک های جبر خطی [۴] توصیف کرد. بعداً، فوزر^۱ دو الگوریتم معروف خود برای محاسبه پایه های گرینر ارائه کرد (مراجع [۵، ۶] را مطالعه کنید). از آنجایی که مطالعه ما بر پایه های گرینر پارامتری، مشهور به دستگاه گرینر^۲، متمرکز است به طور خلاصه، تاریخچه موجود در مورد این موضوع را مرور می کنیم. مفهوم دستگاه های گرینر و همچنین اولین الگوریتم برای محاسبه آنها برای اولین بار توسط وایزفنینگ^۳ در سال ۱۹۹۲ در [۷] معرفی شد. در این سال ها، مطالعات موثر و نتایج بسیاری در محاسبات سیستم های گرینر به دست آمده است [۸-۲۴] که نشان از اهمیت و کاربرد این مفهوم دارد. در نهایت دهقانی روش کارآمد خود را در [۹] معرفی کرد که با استفاده از تکنیک های جبر خطی پارامتری به افزایش فضای پارامتری می پردازد. سیستم های گرینر کاربردهای متعددی در ریاضیات و سایر علوم دارند. به طور خاص، می توان به هندسه جبری [۲۵-۲۹]، جبر خطی پارامتری [۸، ۱۰، ۱۵]، بیوشیمی و بیولوژی [۳۰، ۳۱]، اثبات خودکار برخی قضایای هندسه [۱۶، ۱۹، ۲۵، ۲۸، ۲۹، ۳۲، ۳۳]، شبکه های الکتریکی [۳۴] و غیره اشاره کرد. برای رسیدن به هدف اصلی این تحقیق، ساختار مقاله بصورت زیرتنظیم شده است: در بخش ۱ به بیان مفهوم پایه گرینر و دستگاه گرینر می پردازیم. معرفی مختصری از مساله سینماتیک معکوس در بخش ۲ انجام می شود و چگونگی کاربرد پایه های گرینر برای حل این مساله برای رباتی مسطح ساده توضیح داده می شود. در بخش ۳ به حل مساله سینماتیک معکوس ربات رومین با کمک دستگاه گرینر خواهیم پرداخت.

پایه های گرینر و دستگاه گرینر

مفهوم پایه های گرینر و اولین الگوریتم محاسبه ی آن ها در رساله ی دکترای برونو بوخبرگر تحت راهنمایی ولفگانگ گرینر در سال ۱۹۶۵ معرفی شدند [۳]. در واقع هدف بوخبرگر یافتن نمایش مناسبی برای فضای خارج قسمتی متناظر با یک ایده آل چندجمله ای از بعد صفر بود. این نمایش مناسب، زمینه ساز معرفی پایه ی استاندارد شد که بوخبرگر بعدها به افتخار استاد راهنمایش آن را پایه ی گرینر نامید. الگوریتم بوخبرگر^۴ از پیچیدگی محاسبات بالایی برخوردار بود. وی توانست در سال ۱۹۷۹ با ارائه ی دو محک، الگوریتم خود را قدرتمندتر و سریعتر نماید [۳]. دانیل لازارد در سال ۱۹۸۳ در [۴] با استفاده از جبر خطی توانست محاسبه ی پایه ی گرینر را بهبود بخشد که این خود نقطه ی عطفی در زمینه ی بهبود محاسبه ی پایه ی گرینر است. در نهایت فوزر دو الگوریتم مشهور خود یعنی F_4 و F_5 را ارائه کرد [۵، ۶] که الگوریتم F_5 سریع ترین الگوریتم محاسبه ی پایه ی گرینر تاکنون است. در ادامه این پایه ها را مورد مطالعه قرار می دهیم. برای این منظور در ابتدا حلقه ی چندجمله ای های چند متغیره را بیان و سپس یک ترتیب تک جمله ای را معرفی می کنیم. یک تک جمله ای بر حسب متغیرهای x_1, \dots, x_n حاصل ضربی از متغیرهای تواندار به شکل $x^\alpha = x_1^{\alpha_1} \dots x_n^{\alpha_n}$ است به قسمی که $\alpha = (\alpha_1, \dots, \alpha_n)$ برداری از اعداد صحیح نامنفی باشد. اگر بردار $\alpha = (0, \dots, 0)$ آنگاه $x^\alpha = 1$ ، همچنین درجه ی x^α را با

^۱ Faugère

^۲ Gröbner system

^۳ Weispfenning

^۴ Buchberger's algorithm

نماد $|\alpha| = \sum_{i=1}^n a_i$ به صورت $|\alpha|$ تعریف می‌کنیم. فرض کنیم \mathbb{K} یک میدان باشد. هر چندجمله‌ای f بر حسب متغیرهای x_1, \dots, x_n ترکیب خطی تعدادی متناهی از تک جمله‌ای‌ها با ضرایب در \mathbb{K} است که به شکل $f = \sum_{\alpha \in A} a_\alpha x^\alpha$ می‌باشد و در آن $A \subset \mathbb{Z}_{\geq 0}^n$ زیر مجموعه‌ای متناهی است و $a_\alpha \in \mathbb{K}$ ها ضرایب تک جمله‌ای‌های x^α در f هستند. همچنین، درجه‌ی چندجمله‌ای f را با نماد $\deg(f)$ نمایش داده و به صورت زیر تعریف می‌کنیم:

$$\deg(f) = \max\{|\alpha| \mid a \in A, a_\alpha \neq 0\}$$

مجموعه متشکل از همه‌ی چندجمله‌ای‌ها بر حسب متغیرهای x_1, \dots, x_n با ضرایب در \mathbb{K} به همراه دو عمل جمع و ضرب چندجمله‌ای‌ها تشکیل حلقه‌ای می‌دهند که به حلقه‌ی چندجمله‌ای‌های چند متغیره مرسوم است و با نماد $\mathfrak{R} = \mathbb{K}[x_1, \dots, x_n]$ نمایش داده می‌شود. بر خلاف حلقه‌ی چندجمله‌ای‌های یک متغیره که درجه‌ی تک جمله‌ای‌ها نقش تعیین کننده‌ای در مقایسه‌ی آن‌ها دارد، در حلقه‌ی چندجمله‌ای‌های چند متغیره نیاز به تعریف نوعی ترتیب داریم: یک ترتیب کلی $<$ روی $\mathbb{Z}_{\geq 0}^n$ را یک ترتیب تک جمله‌ای گوئیم^۱ هرگاه به ازای هر $\alpha, \beta, \gamma \in \mathbb{Z}_{\geq 0}^n$:

$$1. \text{ اگر } \alpha < \beta \text{ آنگاه } \alpha + \gamma < \beta + \gamma$$

$$2. \text{ ترتیب } < \text{ روی } \mathbb{Z}_{\geq 0}^n \text{ خوش ترتیب باشد.}$$

فرض کنیم $\text{MI}(\mathfrak{R})$ مجموعه‌ی تمام تک جمله‌ای‌های حلقه‌ی \mathfrak{R} باشد. با استفاده از تناظر یک به یک زیر یک ترتیب تک جمله‌ای روی $\mathbb{Z}_{\geq 0}^n$ به یک ترتیب تک جمله‌ای روی \mathfrak{R} تبدیل می‌شود:

$$\text{MI}(\mathfrak{R}) \rightarrow \mathbb{Z}_{\geq 0}^n$$

$$x_1^{\alpha_1} \dots x_n^{\alpha_n} \rightarrow (\alpha_1, \dots, \alpha_n)$$

بنابر این تناظر، به ازای دو تک جمله‌ای $x^\alpha, x^\beta \in \mathfrak{R}$ اگر $\alpha < \beta$ آنگاه تعریف می‌کنیم $x^\alpha < x^\beta$ از مهم‌ترین ترتیب‌های تک جمله‌ای می‌توان به ترتیب الفبایی^۲ اشاره کرد که در آن گوئیم $\alpha <_{lex} \beta$ هرگاه اولین مؤلفه‌ی ناصفر سمت چپ $\alpha - \beta$ مثبت باشد.

فرض کنید $\mathfrak{R} = \mathbb{K}[x_1, \dots, x_n]$ و $<$ یک ترتیب تک جمله‌ای روی \mathfrak{R} باشد. چندجمله‌ای

$$f = a_\alpha x^\alpha + \dots + a_\gamma x^\gamma$$

را که در آن $x^\gamma < \dots < x^\alpha$ و $a_\alpha \neq 0$ در نظر می‌گیریم. منظور از تک جمله‌ای پیشرو^۳، ضریب پیشرو^۴ و جمله‌ی پیشروی^۵ f نسبت به $<$ به ترتیب $a_\alpha x^\alpha$ و $a_\alpha x^\gamma$ است که با نمادهای $\text{LC}_<(f)$ ، $\text{LM}_<(f)$ و $\text{LT}_<(f)$ نمایش داده می‌شود. همچنین ایده‌آل $I \subset \mathbb{K}[x_1, \dots, x_n]$ را ایده‌آلی تک جمله‌ای نامیم هرگاه توسط یک مجموعه (متناهی یا نامتناهی) از تک جمله‌ای‌ها تولید شود. فرض کنیم $I \subset \mathfrak{R} = \mathbb{K}[x_1, \dots, x_n]$

^۱ Monomial order

^۲ Lexicographic order

^۳ Leading monomial

^۴ Leading coefficient

^۵ Leading term

ایده‌آلی در حلقه‌ی چندجمله‌ای‌ها و \prec ترتیب تک‌جمله‌ای روی R باشد. در این صورت ایده‌آل پیشروی^۱ I را نسبت به \prec به صورت زیر تعریف می‌کنیم:

$$LT_{\prec}(I) = \langle LT_{\prec}(f) \mid f \in I \rangle$$

همان‌طور که مشخص است $LT_{\prec}(I)$ ایده‌آلی تک‌جمله‌ای است. در سال ۱۹۱۳ دیکسون^۲ ثابت کرد که هر ایده‌آل تک‌جمله‌ای دارای یک مجموعه مولد متناهی است (لم دیکسون^۳) پس چندجمله‌ای‌های $g_1, \dots, g_k \in I$ موجودند به‌قسمی که

$$LT_{\prec}(I) = \langle LT_{\prec}(g_1), \dots, LT_{\prec}(g_k) \rangle$$

به مجموعه‌ی ذکر شده یعنی $G = \{g_1, \dots, g_k\} \subset I$ یک پایه‌ی گربرنر I نسبت به \prec می‌گوییم. باید توجه کنیم که پایه‌ی گربرنر یک ایده‌آل یکتا نیست و به ترتیب تک‌جمله‌ای \prec وابسته است. به عنوان مثال اگر $I = \langle y^2 + x, y^3 + 1 \rangle$ ایده‌آلی در $\mathbb{K}[x, y]$ باشد آنگاه با استفاده از دستور BASIS در نرم افزار مپیل مجموعه‌ی $G_1 = \langle y^2 + x, y^3 + 1 \rangle$ یک پایه‌ی گربرنر برای I نسبت به ترتیب $x \prec_{lex} y$ است در حالی که مجموعه‌ی $G_2 = \langle y^2 + x, xy - 1, x^2 + y \rangle$ یک پایه‌ی گربرنر I نسبت به ترتیب $x \prec_{drl} y$ می‌باشد.

یکی از مهم‌ترین کاربردهای پایه‌های گربرنر حل دستگاه‌های چندجمله‌ای در $\mathbb{K}[x_1, \dots, x_n] = \mathfrak{R}$ می‌باشد که به اختصار به کمک یک مثال توضیح می‌دهیم. فرض کنیم $I = \langle f_1, \dots, f_s \rangle$ ایده‌آلی در $\mathfrak{R} = \mathbb{K}[x_1, \dots, x_n]$ باشد. منظور از چندگونای^۴ ایده‌آل I که آن را با $V(I)$ یا $V(f_1, \dots, f_s)$ نمایش می‌دهیم زیرمجموعه‌ی زیر از \mathbb{K}^n است:

$$V(I) = \{(a_1, \dots, a_n) \in \mathbb{K}^n \mid f_i(a_1, \dots, a_n) = 0, \forall i = 1, \dots, s\} \subset \mathbb{K}^n.$$

فرض کنیم G پایه‌ی گربرنر ایده‌آل I نسبت به یک ترتیب تک‌جمله‌ای باشد. در این صورت

$$V(I) = \emptyset \iff 1 \in G$$

به کمک این رابطه دو شرطی می‌توانیم به این سؤال پاسخ دهیم که آیا یک دستگاه چندجمله‌ای دارای جواب هست یا خیر. در صورت مثبت بودن پاسخ این سؤال، به‌وسیله پایه گربرنر می‌توانیم همه جواب‌های یک دستگاه‌های چندجمله‌ای را بیابیم.

مثال ۱. دستگاه چندجمله‌ای زیر را در نظر بگیرید.

$$\begin{cases} x^2 - 4y = 12z \\ z^2 - 1 = 0 \\ yz - 1 = 0 \end{cases} \quad (1)$$

^۱ Leading ideal

^۲ Dickson

^۳ Dickson's lemma

^۴ Variety

برای پاسخ به این سؤال که آیا دستگاه (۱) دارای جواب هست و در صورت مثبت بودن، آیا تعداد ریشه‌ها متناهی است ابتدا ایده‌آل $I = \langle x^2 - 4y - 12, z^2 - 1, yz - 1 \rangle \subset \mathbb{C}[x, y, z]$ را تشکیل می‌دهیم و پایه‌ی گربنر I را نسبت به ترتیب $x \prec_{lex} y \prec_{lex} z$ محاسبه می‌کنیم که برابر $G = \{z^2 - 1, y - z, x^2 - 16z\}$ است. چون $G \neq \{1\}$ پس نتیجه می‌شود $V(I) \neq \emptyset$ و لذا دستگاه (۱) در \mathbb{C} دارای جواب است. از طرفی چون ایده‌آل I صفر بعدی است پس دارای تعداد متناهی ریشه است که به راحتی با استفاده از G به صورت زیر به دست می‌آیند:

$$z^2 - 1 = 0 \Rightarrow z = \pm 1$$

$$y - z = 0 \Rightarrow y = z \Rightarrow y = \pm 1$$

$$x^2 - 16z = 0 \Rightarrow x = \pm 4 \text{ or } x = \pm 4i$$

پس همه ریشه‌های دستگاه چندجمله‌ای‌های (۱)، یعنی همان چندگونای $V(I)$ عبارتست از

$$V(I) = \{(4, 1, 1), (-4, 1, 1), (4i, 1, 1), (-4i, 1, 1)\}$$

تا کنون پایه‌ی گربنر ایده‌آل I در حلقه‌ی $\mathfrak{R} = \mathbb{K}[x_1, \dots, x_n]$ را نسبت به یک ترتیب تک جمله‌ای مطالعه کردیم که در آن ضرایب چندجمله‌ای‌ها در میدان دلخواه \mathbb{K} قرار داشتند. در ادامه قصد داریم به مطالعه‌ی ایده‌آل‌های چندجمله‌ای بپردازیم که در آن ضرایب چندجمله‌ای‌ها خود چندجمله‌ای‌هایی بر حسب تعدادی پارامتر می‌باشند سپس پایه‌ی گربنر این گونه ایده‌آل‌ها را معرفی و محاسبه می‌کنیم که به پایه‌ی گربنر پارامتری مشهورند و با توجه به ساختاری که دارند دستگاه گربنر نامیده می‌شوند. انگیزه‌ی اصلی پیدایش مفهوم دستگاه گربنر را می‌توان مسئله‌ی زیر دانست که توسط بایر^۱ در سال ۱۹۹۱ در مطرح شد:

فرض کنیم R و R' دو حلقه‌ی جابجایی نوتری^۲ یک‌دار باشند و $\sigma: R \rightarrow R'$ یک هم‌ریختی^۳ حلقه‌ای باشد. اگر

$$I \subset \mathbb{C}[x_1, \dots, x_n] \text{ و } \sigma: R[x_1, \dots, x_n] \rightarrow R'[x_1, \dots, x_n] \text{ توسیع طبیعی } \sigma \text{ و } G \text{ یک پایه‌ی گربنر ایده‌آل } I \subset \mathbb{C}[x_1, \dots, x_n]$$

$R[x_1, \dots, x_n]$ باشد، آنگاه چه هنگام $\sigma(G)$ یک پایه‌ی گربنر ایده‌آل $\langle \sigma(I) \rangle$ در $R'[x_1, \dots, x_n]$ است؟

حاصل تلاش محققین برای پاسخ به این سوال، منجر به پیدایش مفهومی با عنوان دستگاه گربنر شد. این مفهوم برای اولین بار در سال ۱۹۹۲ توسط وایزفنینگ در [۷] معرفی شد (به‌طور هم‌زمان و جداگانه سیت^۴ در [۴] نیز به مطالعه‌ی دستگاه‌های چندجمله‌ای پارامتری پرداخت). وی در این مقاله اولین الگوریتم محاسبه‌ی دستگاه گربنر را ارائه کرد. دلزمن^۵ و اشتورم^۶ در سال ۱۹۹۷ در [۴] اولین اجرای الگوریتم وایزفنینگ را ارائه کردند. در همین سال کالک برنر^۷ در [۳۵] پایایی پایه‌های گربنر تحت تاثیر یک هم‌ریختی خاص به نام تخصیص^۸ را مورد مطالعه قرار داد که سال‌ها بعد زمینه‌ی پیدایش الگوریتم‌های سریع‌تر برای محاسبه‌ی این دو مفهوم را فراهم کرد. در سال

^۱ Bayer

^۲ Noetherian ring

^۳ Homeomorphism

^۴ Sit

^۵ Dolzmann

^۶ Sturm

^۷ Kalkbrener

^۸ Specialization

۲۰۰۲ مونتس^۱ در [۲۰] با استفاده از ایده‌ی طراحی الگوریتم وایزفنینگ در [۷]، اولین الگوریتم سریع و کاربردی محاسبه‌ی دستگاه گربنر را ارائه کرد (همچنین [۱۲]). یک سال بعد وایزفنینگ در [۴] پایه‌ی گربنر فراگیر متعارف را معرفی نمود. در سال ۲۰۰۶ سوزوکی^۲ و ساتو^۳ در [۲۳] با استفاده از نتایج بسیار مهم و مفید کالک برنر در [۳۵] توانستند شیوه‌ای جدید برای محاسبه‌ی دستگاه گربنر و پایه‌ی گربنر فراگیر ارائه کنند. در سال ۲۰۱۰ کاپور^۴ و همکارانش در [۱۶] با استفاده از الگوریتم ساتو و سوزوکی و همچنین ایده‌های وایزفنینگ در [۷] توانستند سریع‌ترین الگوریتم محاسبه‌ی دستگاه گربنر را طراحی کنند. در ادامه این مفهوم را معرفی می‌کنیم.

فرض کنیم $S = \mathbb{K}[a][x]$ حلقه‌ی چندجمله‌ای‌های پارامتری باشد که در آن $x = x_1, \dots, x_n$ دنباله‌ای از متغیرها و $a = a_1, \dots, a_m$ دنباله‌ای از پارامترها است. هر تک‌جمله‌ای در S به شکل $(\prod a_i)^\delta (\prod x_j)^\gamma$ است که در آن $\delta \in \mathbb{N}^m$ و $\gamma \in \mathbb{N}^n$ و برای سادگی به صورت $a^\delta x^\gamma$ نوشته می‌شود. برای مقایسه‌ی تک‌جمله‌ای‌های موجود در S نیاز به معرفی یک ترتیب تک‌جمله‌ای است: اگر $x <_x$ یک ترتیب تک‌جمله‌ای روی متغیرها و $a <_a$ یک ترتیب تک‌جمله‌ای روی پارامترها باشد آنگاه ترتیب تک‌جمله‌ای $x, a <_{x,a}$ عبارتست از

$$a^\delta <_a a^\beta \text{ و } x^\gamma = x^\alpha \text{ یا } x^\gamma <_x x^\alpha \text{ اگر } x^\gamma a^\delta <_{x,a} x^\alpha a^\beta$$

که در آن $\delta, \beta \in \mathbb{N}^m$ و $\alpha \in \mathbb{N}^n$

مثال ۲. اگر $x: y <_{lex} x$ ترتیب روی متغیرها و $a: b <_{ari} a$ ترتیب روی پارامترها باشد، آنگاه $a^2 x^2 y <_{x,a} a^2 b x^2 y$ و $a^2 b x y <_{x,a} a x^2$

در حلقه‌ی چندجمله‌ای‌ها با ضرایب پارامتری با سه ترتیب تک‌جمله‌ای $<_a$ ، $<_x$ و $<_{x,a}$ سروکار داریم. بنابراین برای مشخص شدن مفاهیمی همچون تک‌جمله‌ای پیشرو، ضریب پیشرو و ... از اندیس استفاده می‌کنیم. منظور از یک تخصیص، همریختی $\sigma: \mathbb{K}[a] \rightarrow \overline{\mathbb{K}}$ است که در آن $\overline{\mathbb{K}}$ بستر جبری میدان \mathbb{K} می‌باشد. توسع طبیعی این همریختی به صورت زیر است $\sigma: \mathbb{K}[a][x] \rightarrow \overline{\mathbb{K}}[x]$

که به‌ازای هر $f(a, x) \in S$ و $a_o = (t_1, \dots, t_m) \in \mathbb{K}^m$ داریم $\sigma(f) = f(a_o, x) \in \overline{\mathbb{K}}[x]$

تعریف ۳. فرض کنیم $F, G_i \subset S$ و $(N_i, W_i) \subset \mathbb{K}[a] \times \mathbb{K}[a]$ به‌ازای $i = 1, \dots, l$ مجموعه‌ی سه-تایی‌های $\mathcal{G} = \{(N_i, W_i, G_i)\}_{i=1}^l$ یک دستگاه گربنر ایده‌آل $\langle f \rangle$ روی $\mathbb{A} \subset \overline{\mathbb{K}}^m$ نسبت به ترتیب تک-جمله‌ای $<_{x,a}$ نامیده می‌شود هرگاه به‌ازای هر i داشته باشیم:

- مجموعه‌ی $\sigma(G_i) \subset \overline{\mathbb{K}}[x]$ یک پایه گربنر $\langle \sigma(F) \rangle$ نسبت به $<_x$ باشد که $\sigma: \mathbb{K}[a] \rightarrow \overline{\mathbb{K}}$ تخصیص دلخواهی است و در (N_i, W_i) صدق می‌کند.
- $\bigcup_{i=1}^l \mathbb{V}(N_i) \setminus \mathbb{V}(W_i) = \mathbb{A}$

^۱ Montes

^۲ Suzuki

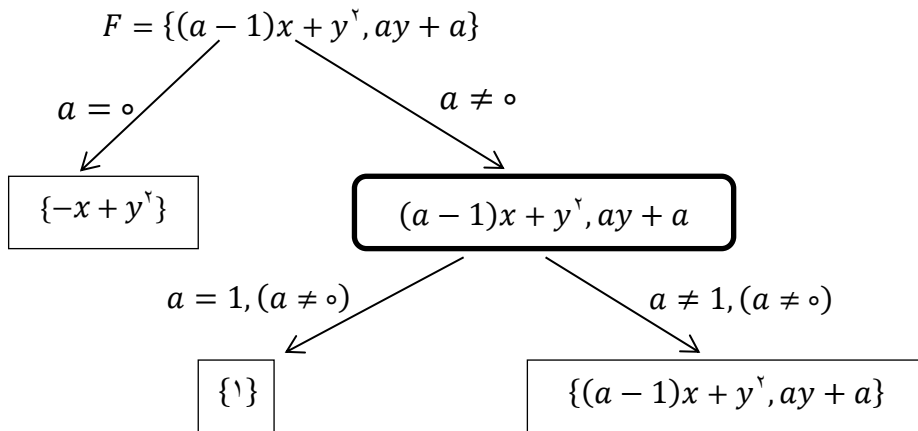
^۳ Sato

^۴ Kapur

اگر $A = \overline{\mathbb{K}}^m$ آنگاه به صورت خلاصه \mathcal{G} را یک دستگاه گرینر $\langle F \rangle$ می‌نامیم. به هر یک از سه تایی‌های (N_i, W_i, G_i) در \mathcal{G} یک شاخه‌ی دستگاه گرینر می‌گوییم. با توجه به شرط دوم تعریف بالا، N_i را مجموعه‌ی شرایط صفر^۱ و W_i را مجموعه‌ی شرایط ناصفر^۲ می‌نامیم. همچنین زوج (N_i, W_i) را زوج سازگار^۳ می‌گوییم هرگاه $V(N_i) \setminus V(W_i) \neq \emptyset$ و در غیر این صورت آن را ناسازگار می‌نامیم که از دستگاه گرینر قابل حذف می‌باشند. در ادامه مثالی از دستگاه گرینر یک ایده‌آل را بیان می‌کنیم که با الگوریتم GES-GVW-CGS که در میپل اجرا کرده ام [9] محاسبه شده است.

مثال ۴. فرض کنیم $F = \{(a-1)x + y^2, ay + a\} \subset \mathbb{K}[a][x, y]$ که پارامتر a و متغیر x, y هستند. یک دستگاه گرینر $\langle F \rangle$ نسبت به ترتیب تک‌جمله‌ای $x \prec_{lex} y \prec_{lex} a$ عبارتست از

$$\left\{ \begin{array}{lll} (\{a\}, & \{\}, & \{-x + y^2\}) \\ (\{a-1\}, & \{a\}, & \{1\}) \\ (\{\}, & \{a, a-1\}, & \{(a-1)x + y^2, ay + a\}) \end{array} \right.$$



شکل ۳: نمودار درختی دستگاه گرینر ایده‌آل ورودی

حل مسئله سینماتیک معکوس ربات رومین

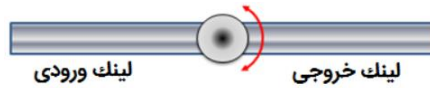
در این بخش ابتدا مسئله‌ی سینماتیک معکوس در رباتیک را معرفی و سپس آن را برای یک ربات ساده دو بعدی حل می‌کنیم [۲۰]. سپس با تعمیم این روش، مساله سینماتیک معکوس ربات سه بعدی رومین را حل نموده و

^۱ Null

^۲ Not null

^۳ Consistent pair

تمام حالات تکین آنرا بصورت کاملاً اتومات محاسبه می‌کنیم. ربات‌ها متناسب با استفاده‌ای که از آن‌ها می‌شود اشکال مختلفی دارند و تقریباً تمامی آنها دارای بازوهایی می‌باشند. بازوی اکثر ربات‌های مورد مطالعه ما شامل دو نوع مفصل گردشی^۱ و مفصل منشوری^۲ می‌باشد:



شکل ۴: مفصل گردشی ربات

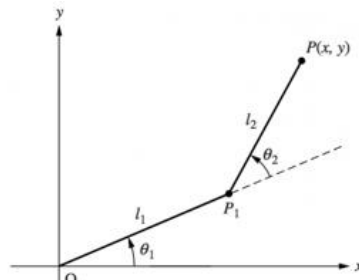


شکل ۵: مفصل منشوری ربات

یک مفصل گردشی امکان دوران یک قطعه از بازو را در صفحه فراهم می‌کند. همچنین یک مفصل منشوری امکان لغزش یا انتقال یک قسمت از بازوی ربات را به داخل قطعه‌ی دیگر فراهم می‌کند. در حالت کلی حوزه‌ی گردش قسمتی از بازو که متصل به مفصل گردشی مسطح است، یک دایره‌ی S^1 می‌باشد. (البته در برخی موارد این حوزه زیر مجموعه‌ای از S^1 است.) همچنین دامنه‌ی انتقال یا لغزش یک قسمت از بازوی متصل به مفصل منشوری، بازه‌ای مانند I می‌باشد که طول این بازه معمولاً کوچکتر از مساوی طول آن قطعه از بازو است. اکنون فرض کنیم بازوی رباتی شامل m مفصل گردشی مسطح و p مفصل منشوری باشد. اگر موقعیت هر مفصل را بتوانیم به صورت مستقل مشخص کنیم آنگاه موقعیت کلی مجموعه‌ی همه‌ی مفاصل با حاصل ضرب دکارتی زیر مشخص می‌شود:

$$\tau = S_1^1 \times \dots \times S_n^1 \times I_1 \times \dots \times I_p$$

τ را فضای مفصل ربات^۳ می‌نامیم. برای توصیف پیکربندی دست ربات مسطح به صورت زیر عمل می‌کنیم:



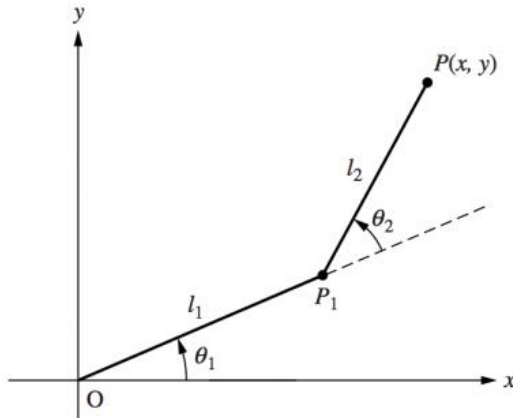
شکل ۶: مدل‌سازی ربات با دو بازو

^۱ Revolute joints

^۲ Prismatic joints

^۳ Robot joint space

یک دستگاه مختصات دکارتی ثابت را در صفحه در نظر می‌گیریم. موقعیت‌های قابل دسترسی ربات را با نقاط $(a, b) \in U \subset \mathbb{R}^2$ نمایش می‌دهیم. همچنین جهت دست ربات را بردار یکه‌ی u در نظر می‌گیریم. پس جهات قابل دسترسی با بردار یکه‌ی u در $V = S^1$ پارامتری می‌شوند. $C = U \times V$ را فضای پیکربندی دست ربات می‌نامیم. چون قطعات بازوی ربات سفت و تانשו در نظر گرفته می‌شوند هر مجموعه از موقعیت‌های مفاصل، مکان دست را در یک حالت یکتا با جهت منحصر بفرد بیان می‌کنند. حال می‌توانیم تابع $f: \tau \rightarrow C$ را در نظر بگیریم. اکنون یک مسئله‌ی اساسی می‌تواند این‌گونه مطرح شود:



شکل ۷: سینماتیک معکوس ربات دو بعدی

فرض کنیم $c \in C$. آیا می‌توانیم $i \in \tau$ را طوری معین کنیم که $f(i) = c$ ؟

سوال فوق مسئله‌ی سینماتیک معکوس نام دارد. اکنون این مسئله را با استفاده از دستگاه گرنر برای یک ربات ساده حل می‌کنیم. این ربات دارای دو بازو به طول l_1 و l_2 است که بازوی l_1 با یک مفصل گردشی مسطح به مبدأ مختصات قائم متصل می‌باشد. همچنین l_1 و l_2 با مفصل گردشی مسطح به یکدیگر متصل شده‌اند. برای حل این مسئله به دو دستگاه مختصات قائم (x_1, y_1) و (x_{i+1}, y_{i+1}) نیاز داریم. دستگاه مختصات (x_1, y_1) ، دستگاه مختصات سرتاسری و (x_{i+1}, y_{i+1}) دستگاه مختصات موضعی نامیده می‌شود. در هر مفصل i ، یک دستگاه مختصات (x_{i+1}, y_{i+1}) را به‌صورت زیر معرفی می‌کنیم:

مبدأ دستگاه مختصات (x_{i+1}, y_{i+1}) در مفصل i قرار می‌گیرد. قسمت مثبت محور x_{i+1} امتداد بازوی $i + 1$ می‌باشد. فرض کنیم نقطه‌ی p در دستگاه (x_{i+1}, y_{i+1}) دارای مختصات (a_i, b_i) باشد. اولین هدف ما تعیین مختصات نقطه‌ی p در دستگاه (x_{i+1}, y_{i+1}) است. فرض می‌کنیم زاویه‌ی بین محور x_i و محور x_{i+1} در خلاف جهت حرکت عقربه‌های ساعت باشد. مختصات مفصل i در دستگاه مختصات (x_i, y_i) برابر $(l_i, 0)$ می‌باشد

که l_i طول بازوی متصل به مفصل i است. اکنون قصد داریم مبدأ دو دستگاه مختصات (x_1, y_1) و (x_{i+1}, y_{i+1}) را بر هم منطبق کنیم. برای این کار محور x_{i+1} را به اندازه θ_i دوران می‌دهیم. سپس با بردار انتقال $(l_i, 0)$ دو مبدأ بر هم منطبق می‌شوند. پس داریم:

$$\begin{pmatrix} a_i \\ b_i \end{pmatrix} = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{pmatrix} \begin{pmatrix} a_{i+1} \\ b_{i+1} \end{pmatrix} + \begin{pmatrix} l_i \\ 0 \end{pmatrix}$$

که در آن $\begin{pmatrix} \cos \theta_i & -\sin \theta_i \\ \sin \theta_i & \cos \theta_i \end{pmatrix}$ همان ماتریس دوران متناسب با زاویه θ_i می‌باشد. معمولاً رابطه‌ی فوق را به صورت زیر نیز نمایش می‌دهند:

$$\begin{pmatrix} a_i \\ b_i \\ 1 \end{pmatrix} = \begin{pmatrix} \cos \theta_i & -\sin \theta_i & l_i \\ \sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} a_{i+1} \\ b_{i+1} \\ 1 \end{pmatrix} = A_i \begin{pmatrix} a_{i+1} \\ b_{i+1} \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = A_1 A_2 \begin{pmatrix} x_3 \\ y_3 \\ 1 \end{pmatrix} \Rightarrow$$

$$\begin{pmatrix} x_1 \\ y_1 \\ 1 \end{pmatrix} = \begin{pmatrix} \cos(\theta_1 + \theta_2) & -\sin(\theta_1 + \theta_2) & l_2 \cos(\theta_1 + \theta_2) + l_1 \cos \theta_1 \\ \sin(\theta_1 + \theta_2) & \cos(\theta_1 + \theta_2) & l_2 \sin(\theta_1 + \theta_2) + l_1 \sin \theta_1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_3 \\ y_3 \\ 1 \end{pmatrix}$$

قرار می‌دهیم $c_i = \cos \theta_i$, $s_i = \sin \theta_i$, $x_1 = x$ و $y_1 = y$ همچنین داریم:

$$\cos(\theta_1 + \theta_2) = \cos \theta_1 \cos \theta_2 - \sin \theta_1 \sin \theta_2 = c_1 c_2 - s_1 s_2$$

$$\sin(\theta_1 + \theta_2) = \sin \theta_1 \cos \theta_2 + \cos \theta_1 \sin \theta_2 = s_1 c_2 + c_1 s_2 \Rightarrow$$

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} c_1 c_2 - s_1 s_2 & -(s_1 c_2 + c_1 s_2) & l_2(c_1 c_2 - s_1 s_2) + l_1 c_1 \\ s_1 c_2 + c_1 s_2 & c_1 c_2 - s_1 s_2 & l_2(s_1 c_2 + c_1 s_2) + l_1 s_1 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_3 \\ y_3 \\ 1 \end{pmatrix}$$

چون $(x_3, y_3) = (0, 0)$ نتیجه می‌گیریم:

$$\begin{aligned} x &= l_2(c_1 c_2 - s_1 s_2) + l_1 c_1 \\ y &= l_2(s_1 c_2 + c_1 s_2) + l_1 s_1 \end{aligned} \quad (2)$$

از طرفی می‌دانیم $\cos^2 \theta + \sin^2 \theta - 1 = 0$ بنابراین داریم:

$$\begin{aligned} c_1^2 + s_1^2 - 1 &= 0 \\ c_2^2 + s_2^2 - 1 &= 0 \end{aligned} \tag{۳}$$

اکنون ایده آل تولید شده توسط چندجمله‌ای‌های (۲) و (۳) را در نظر می‌گیریم:

$$F = \{l_2(c_1c_2 - s_1s_2) + l_1c_1 - x, l_2(s_1c_2 - c_1s_2) + l_1s_1 - y, c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1\}$$

جدول ۱: دستگاه گربنر ربات دو بعدی

Parametric Constraints	Gröbner Bases
$[, [-4xl_1^2l_2^2(x^2 + y^2)(l_1^2 + l_2^2 - x^2 - y^2)]$	$[4l_1^2l_2^2s_2^2 + l_1^4 - 2l_1^2l_2^2 - 2l_1^2x^2 - 2l_1^2y^2 + l_2^4 - 2l_2^2x^2 - 2l_2^2y^2 + x^4 + 2x^2y^2 + y^4, 2l_1l_2s_2^2 - c_2l_1^2 - c_2l_2^2 + c_2x^2 + c_2y^2 - 2l_1l_2, -c_2l_2y + l_2s_2x + s_1x^2 + s_1y^2 - l_1y, c_1x - c_2l_2 + s_1y - l_1]$
$[x], [l_1, l_2, y, -l_1^2 - l_2^2 + y^2]$	$[4l_1^2l_2^2s_2^2 + l_1^4 - 2l_1^2l_2^2 - 2l_1^2y^2 + l_2^4 - 2l_2^2y^2 + y^4, 2l_1l_2s_2^2 - c_2l_1^2 - c_2l_2^2 + c_2y^2 - 2l_1l_2, -c_2l_2 + s_1y - l_1, c_1y - l_2s_2]$
$[-l_1^2 + l_2^2, y, x], [l_2]$	$[l_2s_2, c_2l_2 + l_1, c_1^2 + s_1^2 - 1]$
$[l_1, l_2, y, x], [1]$	$[c_2^2 + s_2^2 - 1, c_1^2 + s_1^2 - 1]$
$[l_1^2l_2^2, -l_1^3l_2 - l_1l_2^3 + l_1l_2y^2, l_1^4 - 2l_1^2y^2 + l_2^4 - 2l_2^2y^2 + y^4, x], [-l_1^4 + l_1^2y^2 - l_2^4 + l_2^2y^2]$	$[2l_1l_2s_2^2 - c_2l_1^2 - c_2l_2^2 + c_2y^2 - 2l_1l_2, -c_2l_2 + s_1y - l_1, c_1y - l_2s_2]$
$[l_1l_2, -l_1^2 - l_2^2 + y^2, x], [l_1^4 + l_2^4]$	$[c_2^2 + s_2^2 - 1, -c_2l_2 + s_1y - l_1, c_1y - l_2s_2]$
$[-l_1^2 - l_2^2 + y^2, x], [l_1, l_2, l_1^2 + l_2^2]$	$[l_1l_2s_2^2 - l_1l_2, l_1l_2c_2, -c_2l_2 + s_1y - l_1, c_1y - l_2s_2]$
$[l_1l_2, -l_1^2 - l_2^2 + x^2 + y^2], [l_2, -l_2 + y, l_2 + y]$	$[c_2^2 + s_2^2 - 1, -c_2l_2y + l_2^2s_1 + l_2s_2x, c_1x - c_2l_2 + s_1y - l_1]$
$[l_1l_2, l_2^2, l_2y, l_2x, -l_1^2 + x^2 + y^2], [l_1, x, -l_1 + y, l_1 + y]$	$[c_2^2 + s_2^2 - 1, l_1s_1x - xy, c_1x - c_2l_2 + s_1y - l_1]$
$[x^2 + y^2], [l_1, l_2, y, -l_1 + l_2, l_1 + l_2]$	$[2l_1l_2s_2x - l_1^2y + l_2^2y, c_2l_2x + l_2s_2y + l_1x, -l_1^2l_2s_2 - 2l_1^2s_1x + l_2^2s_2 + 2l_2^2s_1x + 2l_1xy, c_1x - c_2l_2 + s_1y - l_1]$
$[-l_1^2 - l_2^2 + x^2 + y^2], [l_1, l_2, l_1^2 + l_2^2, -l_1^2 - l_2^2 + y^2]$	$[l_1l_2s_2^2 - l_1l_2, l_1l_2c_2, -c_2l_2y + l_1^2s_1 + l_2^2s_1 + l_2s_2x - l_1y, c_1x - c_2l_2 + s_1y - l_1]$

دستگاه گربنر مجموعه F نسبت به ترتیب تک‌جمله‌ای $S_2 <_{lex} C_2 <_{lex} S_1 <_{lex} C_1 <_{lex} S_1 <_{lex} C_1 <_{lex} S_1$ روی متغیرها و ترتیب تک‌جمله‌ای $x <_{lex} y <_{lex} l_2 <_{lex} l_1$ روی پارامترها در جدول ۱ آمده است. شایان ذکر است که دستگاه گربنر مجموعه مولد F در [۲۰] محاسبه شده است که در آن طول بازوی اول یعنی l_1 برابر ۱ می‌باشد اما در اینجا l_1 طول یک مفصل منشوری است که بعنوان یک پارامتر غیر ثابت در نظر گرفته می‌شود و نتایج حاصل کلی تر از نتایج مربوط به [۲۰] می‌باشد. این محاسبات در ۱/۹ ثانیه انجام شد و حافظه مورد نیاز ۷۵۶ مگابایت

می‌باشد. همچنین پایه گربنر مجموعه مولد فوق در [۲۶] محاسبه شده است اما پارامترها همچون متغیر همواره ناصفر در نظر گرفته شده است. دستگاه گربنری که در جدول ۱ مشاهده می‌کنید دارای یازده شاخه است. به‌عنوان مثال آخرین عضو خروجی فوق را در نظر می‌گیریم. همان‌طور که می‌بینیم G_{11} بصورت زیر است:

$$[l_1 l_2 s_2^2 - l_1 l_2, l_1 l_2 c_2, -c_2 l_2 y + l_1^2 s_1 + l_2^2 s_1 + l_2 s_2 x - l_1 y, c_1 x - c_2 l_2 + s_1 y - l_1]$$

که پایه‌ی گربنر ایده‌آل F ، متناسب با مجموعه شرایط پارامتری

$$W_{11} = [l_1, l_2, l_1^2 + l_2^2, -l_1^2 - l_2^2 + y^2], N_{11} = [-l_1^2 - l_2^2 + x^2 + y^2]$$

است. این بدان معناست که اگر $-l_1^2 - l_2^2 + x^2 + y^2 = 0$

$$l_1 \neq 0, l_2 \neq 0, l_1^2 + l_2^2 \neq 0, -l_1^2 - l_2^2 + y^2 \neq 0$$

آنگاه دستگاه چندجمله‌ای‌های زیر حاصل می‌شود:

$$\begin{cases} l_1 l_2 s_2^2 - l_1 l_2 & = 0 \\ l_1 l_2 c_2 & = 0 \\ -c_2 l_2 y + l_1^2 s_1 + l_2^2 s_1 + l_2 s_2 x - l_1 y & = 0 \\ c_1 x - c_2 l_2 + s_1 y - l_1 & = 0 \end{cases}$$

و پس از حل دو دسته مجموعه جواب زیر به‌دست می‌آید:

$$\begin{cases} s_2 = 1, & c_2 = 0, & s_1 = \frac{(l_1 y - l_2 x)}{(l_1^2 + l_2^2)}, & c_1 = \frac{(l_1^3 + l_1 l_2^2 - l_1 y^2 + l_2 x y)}{(l_1^2 + l_2^2) x} \end{cases}$$

$$\begin{cases} s_2 = -1, & c_2 = 0, & s_1 = \frac{(l_1 y + l_2 x)}{(l_1^2 + l_2^2)}, & c_1 = \frac{(l_1^3 + l_1 l_2^2 - l_1 y^2 - l_2 x y)}{(l_1^2 + l_2^2) x} \end{cases}$$

با توجه به مجموعه‌های W_{11}, N_{11} به‌عنوان مثال قرار می‌دهیم: $l_1 = 2, l_2 = 4, (x, y) = (2, 4)$

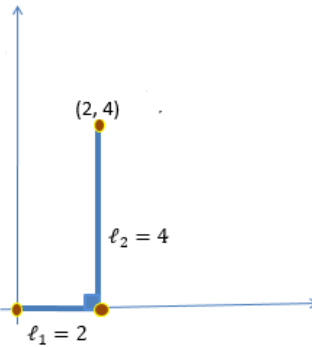
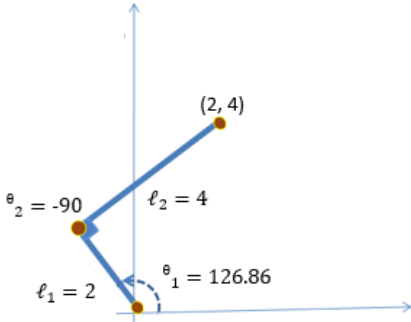
که پس از جایگذاری این مقادیر در دو دسته مجموعه جواب بالا داریم:

$$\begin{cases} s_2 = 1, & c_2 = 0, & s_1 = 0, & c_1 = 1 \end{cases}$$

$$\begin{cases} s_2 = -1, & c_2 = 0, & s_1 = \frac{4}{5}, & c_1 = \frac{-3}{5} \end{cases}$$

بنابراین تمام حالات ممکن برای زاویه مفصل چرخشی بصورت زیر است:

$$\left\{ \begin{array}{ll} \theta_1 = \sin^{-1}(0) = \cos^{-1}(1) = 0^\circ & \text{and } \theta_2 = \sin^{-1}(1) = \cos^{-1}(0) = 90^\circ \\ \theta_1 = \sin^{-1}\left(\frac{4}{5}\right) = \cos^{-1}\left(-\frac{3}{5}\right) = 126.86^\circ & \text{and } \theta_2 = \sin^{-1}(-1) = \cos^{-1}(0) = -90^\circ \end{array} \right.$$



ربات رومین

ربات رومین یک ربات با درجه آزادی سه چرخشی سه بعدی است که دارای سه بازو است که همگی دارای اتصالات چرخشی مسطح و پایه چرخشی هستند [۱]. زاویه θ_1 حول محوری عمود بر سطح زمین می‌چرخد. همچنین زاویه θ_2 (به ترتیب، زاویه θ_3) زاویه اندازه‌گیری شده از صفحه افقی Q1 (به ترتیب، از صفحه Q2) تا بازوی دوم (به ترتیب، تا بازوی سوم) است. برای یک نقطه $(a, b, c) \in \mathbb{R}^3$ بعنوان مختصات دست ربات و با استفاده از ماتریس دوران‌های زیر حول محورهای x, y, z زیر

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{bmatrix}$$

$$R_y(\theta) = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix}$$

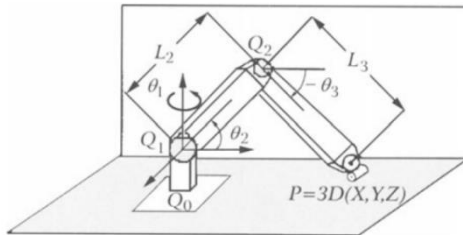
$$R_z(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

دستگاه معادلات زیر حاصل می‌شود که در آن $c_i = \cos \theta_i$ و $s_i = \sin \theta_i$:

$$\begin{cases} -s_1(l_2c_2 + l_3c_3) = a \\ c_1(l_2c_2 + l_3c_3) = b \\ l_2s_2 + l_3s_3 = c \\ s_1^2 + c_1^2 = 1 \\ s_2^2 + c_2^2 = 1 \\ s_3^2 + c_3^2 = 1 \end{cases}$$

از طرفی با توجه به مجموعه مولد بالا چون عبارت $l_2c_2 + l_3c_3$ در هر دو معادله اول دستگاه فوق وجود دارد برای کم شدن پیچیدگی محاسبات متغیر جدید d را برابر $l_2c_2 + l_3c_3$ در نظر می‌گیریم پس داریم:

$$\begin{cases} -s_1d = a \\ c_1d = b \\ l_2c_2 + l_3c_3 = d \\ l_2s_2 + l_3s_3 = c \\ s_1^2 + c_1^2 = 1 \\ s_2^2 + c_2^2 = 1 \\ s_3^2 + c_3^2 = 1 \end{cases}$$



شکل ۸: مدل‌سازی ربات ۳ بعدی رومین

حال اگر بجای $(a, b, c) \in \mathbb{R}^3$ از سه متغیر $(x, y, z) \in \mathbb{R}^3$ به‌عنوان مختصات دست ربات استفاده کنیم مجموعه مولدی با هفت چندجمله‌ای زیر تشکیل می‌شود:

$$F = \{s_1^2 + c_1^2 - 1, s_2^2 + c_2^2 - 1, s_3^2 + c_3^2 - 1, x + ds_1, y - dc_1, l_2c_2 + l_3c_3 - d, l_2s_2 + l_3s_3 - z\}$$

اکنون با کمک الگوریتم GES-GVW-CGS که در نرم افزار میپل اجرا شده است [۹] دستگاه گرنبر F نسبت به ترتیب تک‌جمله‌ای $s_1 <_{lex} c_1 <_{lex} s_2 <_{lex} c_2 <_{lex} s_3 <_{lex} c_3 <_{lex} x <_{lex} y <_{lex} z <_{lex} d$ به ترتیب تک‌جمله‌ای $d <_{lex} z <_{lex} x <_{lex} y <_{lex} l_2 <_{lex} l_1$ روی پارامترها در جدول ۲ آمده است.

جدول ۲: دستگاه گربنر ربات رومین

Parametric Constraints	Gröbner Bases
$[d, z, y, x, l_3, l_2], []$	$[c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1, c_3^2 + s_3^2 - 1]$
$[d, y, x, l_3, l_2], [z]$	$[c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1, -z]$
$[d, y, x, l_3, l_2 - z], [z]$	$[c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1, s_2 z - z, c_2 z, c_3^2 + s_3^2 - 1]$
$[d, y, x, l_3, l_2 + z], [z]$	$[c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1, -s_2 z - z, -c_2 z, c_3^2 + s_3^2 - 1]$
$[d, y, x, l_3], [l_2, l_2 - z, l_2 + z]$	$[c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1, l_2 s_2 - z, l_2 c_2, l_2^2 - z^2]$
$[d, y, x, l_3 - z, l_2], [z]$	$[c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1, s_3 z - z, z c_3]$
$[d, y, x, l_3 + z, l_2], [z]$	$[c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1, -s_3 z - z, -z c_3]$
$[d, y, x, l_2], [l_3, l_3 - z, l_3 + z]$	$[c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1, l_3 s_3 - z, l_3 c_3, -l_3^2 + z^2]$
$[d, z, y, x, l_2 - l_3], [l_3]$	$[c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1, l_3 s_2 + l_3 s_3, c_2 l_3 + c_3 l_3]$
$[d, z, y, x, l_2 + l_3], [l_3]$	$[c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1, -l_3 s_2 + l_3 s_3, -c_2 l_3 + c_3 l_3]$
$[d, z, y, x], [l_2, l_3, l_2 - l_3, l_2 + l_3]$	$[c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1, l_2 s_2 + l_3 s_3, c_2 l_2 + c_3 l_3, l_2^2 - l_3^2]$
$[d, y, x], [l_2, l_3, z]$	$[c_1^2 + s_1^2 - 1, c_2^2 + s_2^2 - 1, l_2 s_2 + l_3 s_3 - z, c_2 l_2 + c_3 l_3, -2l_2 s_2 z + l_2^2 - l_3^2 + z^2]$
$[z, -d^2 + x^2 + y^2, l_3, l_2], [d]$	$[ds_1 + x, c_1 d - y, c_2^2 + s_2^2 - 1, -d]$
$[-d^2 + x^2 + y^2, l_3, l_2], [d, z]$	$[ds_1 + x, c_1 d - y, c_2^2 + s_2^2 - 1, -z]$
$[x^2 + y^2 - d^2, l_3, l_2^2 - d^2 - z^2], [d, l_2]$	$[ds_1 + x, c_1 d - y, c_2^2 + s_2^2 - 1, l_2 s_2 - z, c_2 l_2 - d, c_3^2 + s_3^2 - 1]$
$[-d^2 + x^2 + y^2, l_3], [d, l_2, -d^2 + l_2^2 - z^2]$	$[ds_1 + x, c_1 d - y, c_2^2 + s_2^2 - 1, l_2 s_2 - z, c_2 l_2 - d, -d^2 + l_2^2 - z^2]$
$[x^2 + y^2 - d^2, l_3^2 - d^2 - z^2, l_2], [d, l_3]$	$[ds_1 + x, c_1 d - y, c_2^2 + s_2^2 - 1, l_3 s_3 - z, c_3 l_3 - d]$
$[-d^2 + x^2 + y^2, l_2], [d, l_3, -d^2 + l_3^2 - z^2]$	$[ds_1 + x, c_1 d - y, c_2^2 + s_2^2 - 1, l_3 s_3 - z, c_3 l_3 - d, d^2 - l_3^2 + z^2]$
$[d^2 + z^2, x^2 + y^2 - d^2, l_2 - l_3], [d, l_3, z]$	$[ds_1 + x, c_1 d - y, c_2^2 + s_2^2 - 1, l_3 s_2 + l_3 s_3 - z, c_2 l_3 + c_3 l_3 - d, -2c_2 d l_3 - 2l_3 s_2 z, -4d^2 l_3^2]$
$[d^2 + z^2, x^2 + y^2 - d^2, l_2 + l_3], [d, l_3, z]$	$[ds_1 + x, c_1 d - y, c_2^2 + s_2^2 - 1, -l_3 s_2 + l_3 s_3 - z, -c_2 l_3 + c_3 l_3 - d, 2c_2 d l_3 + 2l_3 s_2 z, -4d^2 l_3^2]$
$[d^2 + z^2, -d^2 + x^2 + y^2], [d, l_2, l_3, z, l_2 - l_3, l_2 + l_3]$	$[ds_1 + x, c_1 d - y, c_2^2 + s_2^2 - 1, l_2 s_2 + l_3 s_3 - z, c_2 l_2 + c_3 l_3 - d, -2c_2 d l_2 - 2l_2 s_2 z + l_2^2 - l_3^2, -4l_3^2 s_2 z + 4l_2 l_3^2 s_2 z - 4d^2 l_3^2 + l_2^2 - 2l_2^2 l_3^2 + l_3^4]$
$[x^2 + y^2 - d^2], [d, l_2, l_3, d^2 + z^2]$	$[ds_1 + x, c_1 d - y, c_2^2 + s_2^2 - 1, l_2 s_2 + l_3 s_3 - z, c_2 l_2 + l_3 - d, -2c_2 d l_3 - 2l_2 s_2 z + d^2 + l_2^2 - l_3^2 + z^2, 4d^2 l_2^2 s_2^2 + 4l_2^2 s_2^2 z^2 - 4d^2 l_2 s_2 z - 4l_3^2 s_2 z + 4l_2 l_3^2 s_2 z - 4l_2 s_2 z^3 + d^4 - 2d^2 l_2^2 - 2d^2 l_3^2 + 2d^2 z^2 + l_2^4 - 2l_2^2 l_3^2 + 2l_3^2 z^2 + l_3^4 - 2l_3^2 z^2 + z^4]$
$[z, l_3, l_2], [d, -d^2 + x^2 + y^2]$	$[ds_1 + x, c_1 d - y, -d^2 + x^2 + y^2]$
$[l_3, l_2], [d, z, -d^2 + x^2 + y^2]$	$[ds_1 + x, c_1 d - y, -d^2 + x^2 + y^2]$
$[l_3], [d, l_2, -d^2 + x^2 + y^2]$	$[ds_1 + x, c_1 d - y, -d^2 + x^2 + y^2]$
$[], [d, l_3, -d^2 + x^2 + y^2]$	$[ds_1 + x, c_1 d - y, -d^2 + x^2 + y^2]$
$[d, z, x, l_3, l_2], [y]$	[1]
$[d, x, l_3, l_2], [y, z]$	[1]
$[d, x, l_3], [l_2, y]$	[1]
$[d, x], [l_3, y]$	[1]
$[d, z, y, l_3, l_2], [x]$	[1]
$[d, y, l_3, l_2], [x, z]$	[1]
$[d, y, l_3], [l_2, x]$	[1]
$[d, y], [l_3, x]$	[1]
$[d, z, l_3, l_2], [x, y]$	[1]
$[d, l_3, l_2], [x, y, z]$	[1]
$[d, l_3], [l_2, x, y]$	[1]
$[d], [l_3, x, y]$	[1]

این دستگاه دارای ۳۸ سه تایی (شاخه) است که در مدت ۵,۶۳ ثانیه محاسبه شده است و مقدار حافظه مورد نیاز برای این محاسبات ۱,۹۲ گیگابایت می‌باشد. اکنون به کمک جدول بالا و بصورت کاملاً خودکار می‌توانیم تمام مقادیر پارامتری ممکن را برای حل مسئله سینماتیک معکوس تحلیل کنیم. برای مثال سطر ۲۲ جدول بالا (ردیف مشخص شده) را در نظر می‌گیریم که در آن $W_{22} = [d, l_2, l_3, d^2 + z^2]$, $N_{22} = [x^2 + y^2 - d^2]$ هستند. این بدان معناست که اگر

$$x^2 + y^2 - d^2 = 0, d \neq 0, l_2 \neq 0, l_3 \neq 0, d^2 + z^2 \neq 0$$

آنگاه لیست زیر پایه گرینر F نسبت به شرایط پارامتری ذکر شده است که آنرا G_{22} می‌نامیم.

$$[ds_1 + x, c_1d - y, c_2^2 + s_2^2 - 1, l_2s_2 + l_3s_3 - z, c_2l_2 + l_3 - d, -2c_2dl_2 - 2l_2s_2z + d^2 + l_2^2 - l_3^2 + z^2, 4d^2l_2^2s_2^2 + 4l_2^2s_2^2z^2 - 4d^2l_2s_2z - 4l_2^3s_2z + 4l_2l_3^2s_2z - 4l_2s_2z^3 + d^4 - 2d^2l_2^2 - 2d^2l_3^2 + 2d^2z^2 + l_2^4 - 2l_2^2l_3^2 + 2l_2^2z^2 + l_3^4 - 2l_3^2z^2 + z^4]$$

حال با توجه به مجموعه شرایط صفر و ناصفر N_{22}, W_{22} به‌عنوان مثال قرار می‌دهیم:

$$l_2 = 3.5, \quad l_3 = 5.5, \quad (x, y, z) = (3, 4, 6) \in \mathbb{R}^3$$

پس از جایگذاری این مقادیر در پایه گرینر G_{22} داریم:

$$[5s_1 + 3, 5c_1 - 4, c_2^2 + s_2^2 - 1, 5.5s_3 + 3.5s_2 - 6, 5.5c_3 + 3.5c_2 - 5, 43 - 35c_2 - 42s_2, 624 - 3612s_2 + 2989s_2^2]$$

$$\begin{cases} 5s_1 + 3 & = 0 \\ 5c_1 - 4 & = 0 \\ c_2^2 + s_2^2 - 1 & = 0 \\ 5.5s_3 + 3.5s_2 - 6 & = 0 \\ 5.5c_3 + 3.5c_2 - 5 & = 0 \\ -35c_2 - 42s_2 + 43 & = 0 \\ 2989s_2^2 - 3612s_2 + 624 & = 0 \end{cases}$$

که منجر به تشکیل دستگاه هفت معادله‌ای زیر می‌شود:

از حل این دستگاه دو دسته مجموعه جواب حاصل می‌شود:

$$\begin{cases} s_3 = 0.9580021314 & c_3 = 0.2867610787 \\ s_2 = 0.2088537935 & c_2 = 0.9779468763 \\ s_1 = -0.6, & c_1 = 0.8 \end{cases}$$

$$\begin{cases} s_3 = 0.4548145601 & c_3 = 0.8905861642 \\ s_2 = 0.9995771198 & c_2 = 0.02907888479 \\ s_1 = -0.6, & c_1 = 0.8 \end{cases}$$

در نتیجه معادل این دو مجموعه جواب، دو حالت زیر برای زاویه بین بازوها به دست می آید:

$$\left\{ \begin{array}{l} \theta_3 = 73.33585444^\circ \quad \text{and} \quad \theta_2 = 12.05519009^\circ \quad \text{and} \quad \theta_1 = -36.86989764^\circ \\ \text{or} \\ \theta_3 = 27.05300333^\circ \quad \text{and} \quad \theta_2 = 88.33366771^\circ \quad \text{and} \quad \theta_1 = -36.86989764^\circ \end{array} \right.$$

در ۱۲ ردیف پایانی جدول ۲، حالت‌های پارامتری را می‌بینیم که دارای پایه گربنر [۲۰] هستند و در این حالت‌ها دستگاه چندجمله‌ای مربوط به مدل جبری ربات رومین فاقد جواب است. در واقع به کمک دستگاه گربنر تمام حالات تکینگی ربات رومین بطور اتومات محاسبه شده‌اند که در جدول ۳ نمایش می‌دهیم:

جدول ۳: حالات تکینگی ربات رومین

Null Sets	Not Null Sets	Gröbner Bases
$[d, z, x, l_3, l_2]$	$[y]$	[1]
$[d, x, l_3, l_2]$	$[y, z]$	[1]
$[d, x, l_3]$	$[l_2, y]$	[1]
$[d, x]$	$[l_3, y]$	[1]
$[d, z, y, l_3, l_2]$	$[x]$	[1]
$[d, y, l_3, l_2]$	$[x, z]$	[1]
$[d, y, l_3]$	$[l_2, x]$	[1]
$[d, y]$	$[l_3, x]$	[1]
$[d, z, l_3, l_2]$	$[x, y]$	[1]
$[d, l_3, l_2]$	$[x, y, z]$	[1]
$[d, l_3]$	$[l_2, x, y]$	[1]
$[d]$	$[l_3, x, y]$	[1]

نتیجه‌گیری

مانند حل مسئله سینماتیک معکوس در ربات‌ها، بسیاری از کاربردهای جالب را می‌توان با استفاده از دستگاه‌های گربنر با نتایج کاملاً رضایت بخش و محاسبات کاملاً خودکار و اتومات بررسی کرد. به عنوان مثال، دستگاه‌های گربنر را می‌توان در پردازش تصویر، رمزنگاری، بیوشیمی و بسیاری از زمینه‌های دیگر که در آن مسائل پیچیده جبری نیاز به تحلیل حالات مختلف شرایط دارند، استفاده نمود. با بکارگیری تکنیک‌های جبر کامپیوتری، محققان و مهندسان می‌توانند محاسبات پیچیده را به راحتی آنالیز کرده و راه‌حلی برای مسائل کاربردی بیابند. قدرت تحلیل دستگاه‌های گربنر آنها را به ابزاری ارزشمند در رشته‌های مختلف علوم و مهندسی تبدیل کرده است. محاسبه و آنالیز خودکار تمام حالات تکینگی ربات رومین مطالعه شده یکی از مصادیق ارزشمند کاربرد تکنیک‌های جبر کامپیوتری در علوم مهندسی است.

تشکر و قدردانی

صمیمانه از پیشنهادات ارزشمند داوران که باعث بهبود کیفیت مقاله شده، قدردانی می‌کنم.

References

- [1] González-López, M., & Recio, T. (1993). The ROMIN inverse geometric model and the dynamic evaluation method, in *Proc. of the 1991 SCAFI Seminar, Computer Algebra in Industry*. Wiley.
- [2] Buchberger, B. (2006). "Bruno Buchberger's PhD thesis 1965: An algorithm for finding the basis elements of the residue class ring of a zero-dimensional polynomial ideal, *J. Symb. Comput.*, 41 (3-4), 475-511. <https://doi.org/10.1016/j.jsc.2005.09.007>
- [3] Buchberger, B. (1979). A criterion for detecting unnecessary reductions in the construction of Gröbner bases, in *International Symposium on Symbolic and Algebraic Manipulation*, 3-21. https://doi.org/10.1007/3-540-09519-5_52
- [4] Lazard, D. (1983). Gröbner bases, Gaussian elimination and resolution of systems of algebraic equations, in *European Conference on Computer Algebra*, 146-156. https://doi.org/10.1007/3-540-12868-9_99
- [5] Faugere, J.-C. (1999). A new efficient algorithm for computing Gröbner bases (F4), *Journal of Pure and Applied Algebra*, 139 (1-3), 61-88. [https://doi.org/10.1016/S0022-4049\(99\)00005-5](https://doi.org/10.1016/S0022-4049(99)00005-5)
- [6] Faugere, J.-C. (2002). A new efficient algorithm for computing Gröbner bases without reduction to zero (F₅), *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, 75-83. <https://doi.org/10.1145/780506.780516>
- [7] Weispfenning, V. (1992). Comprehensive Gröbner bases, *Journal of Symbolic Computation*, 14, (1), 1-29. [doi:10.1016/0747-7171\(92\)90023-W](https://doi.org/10.1016/0747-7171(92)90023-W).
- [8] Dehghani Darmian, M. (2024). Efficient Algorithm for Computing Inverse of Parametric Matrices, *Scientific Annals of Computer Science*, 1, 1-22. [doi:10.47743/SACS.2024.1.1](https://doi.org/10.47743/SACS.2024.1.1).
- [9] Dehghani Darmian, M. (2024). Improvement of an incremental signature-based comprehensive Gröbner system algorithm, *Math. Comput. Sci*, 18 (12). [doi:http://dx.doi.org/10.1007/s11786-024-00587-w](https://dx.doi.org/10.1007/s11786-024-00587-w).
- [10] Dehghani Darmian, M., & Hashemi, A. (2017). Parametric FGLM algorithm, *Journal of Symbolic Computation*, 82, 38-56. [doi:10.1016/j.jsc.2016.12.006](https://doi.org/10.1016/j.jsc.2016.12.006).
- [11] Dehghani Darmian, M., & Hashemi, A. (2024). Parametric F₄ Algorithm, *Iranian Journal of Mathematical Sciences and Informatics*, 19 (4), 117-133. [doi:http://dx.doi.org/10.61186/ijmsi.19.1.117](https://dx.doi.org/10.61186/ijmsi.19.1.117).
- [12] Dehghani Darmian, M., Hashemi, A., & Montes, A. (2011). Erratum to "A new algorithm for discussing Gröbner bases with parameters" *J. Symb. Comput.* 33 (1-2)(2002) 183-208, *J. Symb. Comput.*, 46 (10), 1187-1188. [doi:10.1016/J.JSC.2011.05.002](https://doi.org/10.1016/J.JSC.2011.05.002).

- [13] Hashemi, A., Dehghani Darmian, M., & Barkhordar, M. (2018, July 24-27). Universal Gröbner basis for parametric polynomial ideals, in *Mathematical Software–ICMS 2018: 6th International Conference, South Bend, IN, USA, Proceedings* Springer, 191-199. doi:http://dx.doi.org/10.1007/978-3-319-96418-8_23.
- [14] Hashemi, A., Dehghani Darmian, M., & Barkhordar, M. (2017). Gröbner systems conversion, *Mathematics in Computer Science*, 11, 61-77. doi:[10.1007/S11786-017-0295-3](https://doi.org/10.1007/S11786-017-0295-3).
- [15] Hashemi, A., M.-Alizadeh, B., & Dehghani Darmian, M. (2013). Minimal polynomial systems for parametric matrices, *Linear and Multilinear Algebra*, 61 (2), 265-272. doi:<https://doi.org/10.1080/03081087.2012.670235>.
- [16] Kapur, D., Sun, Y., & Wang, D. (2010). A new algorithm for computing comprehensive Gröbner systems, in *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, 29-36. doi:[10.1145/1837934.1837946](https://doi.org/10.1145/1837934.1837946).
- [17] Kapur, D., Sun, Y., & Wang, D. (2013). An efficient algorithm for computing a comprehensive Gröbner system of a parametric polynomial system, *Journal of Symbolic Computation*, 49, 27-44. <https://doi.org/10.1016/j.jsc.2011.12.015>
- [18] Kapur, D., Sun, Y., & Wang, D. (2013). An efficient method for computing comprehensive Gröbner bases, *Journal of Symbolic Computation. Comput.*, 52, 124-142. <https://doi.org/10.1016/j.jsc.2012.05.015>
- [19] Kapur, D., Sun, Y., Wang, D., & Zhou, J. (2015, July 20–23). The generalized Rabinowitsch trick, in *Applications of Computer Algebra: Kalamata, Greece, Springer*, 219-229. doi:[10.1007/978-3-319-56932-1_14](https://doi.org/10.1007/978-3-319-56932-1_14)
- [20] Montes, A. (2002). A new algorithm for discussing Gröbner bases with parameters, *J. Symb. Comput.*, 33 (2), 183-208. doi:[10.1006/JSCO.2001.0504](https://doi.org/10.1006/JSCO.2001.0504).
- [21] Montes, A., & Wibmer, M. (2010). Gröbner bases for polynomial systems with parameters, *J. Symb. Comput.*, 45 (12), 1391-1425. doi:[10.1016/J.JSC.2010.06.017](https://doi.org/10.1016/J.JSC.2010.06.017).
- [22] Montes, A., & Wibmer, M. (2014, August 5-9). Software for discussing parametric polynomial systems: The Gröbner Cover, in *Mathematical Software–ICMS 2014: 4th International Congress, Seoul, South Korea, Proceedings 4*, 406-413. doi:[10.1007/978-3-662-44199-2_62](https://doi.org/10.1007/978-3-662-44199-2_62)
- [23] Suzuki, A., & Sato, Y. (2006). A simple algorithm to compute comprehensive Gröbner bases using Gröbner bases, in *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, 326-331. doi:[10.1145/1145768.1145821](https://doi.org/10.1145/1145768.1145821).
- [24] Weispfenning, V. (2002). Canonical comprehensive Gröbner bases, in *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, 270- 276. <https://doi.org/10.1145/780506.780541>
- [25] Abánades, M., Botana, F., Montes, A., & Recio, T. (2014, August 5-9). Software using the Gröbner Cover for geometrical loci computation and classification, in *Mathematical Software–ICMS 2014: 4th International Congress, Seoul, South Korea, Proceedings 4*, 492-499. doi:[10.1007/978-3-662-44199-2_74](https://doi.org/10.1007/978-3-662-44199-2_74).
- [26] Cox, D. A., Little, J., & O'Shea, D. (2005). *Using algebraic geometry*. Springer Science & Business Media. <https://doi.org/10.1007/b138611>.
- [27] Dehghani Darmian, M., Hashemi, A. (2010, November 10-11). Classification of cubic curves by Gröbner systems, presented at the 21st Algebra seminar of Iran, Tabriz, Iran.

- [28] Montes, A., & Recio, T. (2006, August 31-September 2). Automatic discovery of geometry theorems using minimal canonical comprehensive Gröbner systems, in *Automated Deduction in Geometry: 6th International Workshop, Pontevedra, Spain*, 113-138. doi: [10.5555/1784950.1784958](https://doi.org/10.5555/1784950.1784958).
- [29] Suzuki, A., & Sato, Y. (2009). Computation of inverses in residue class rings of parametric polynomial ideal, in *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, 311-316. <https://doi.org/10.1145/1576702.15767>
- [30] Dimitrova, E. S., Jarrah, A. S., Laubenbacher, R., & Stigler, B. (2007). A Gröbner fan method for biochemical network modeling, in *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, 122-126. <https://doi.org/10.1145/1277548.127756>
- [31] Laubenbacher, R., & Stigler, B. (2004). A computational algebra approach to the reverse engineering of gene regulatory networks, *Journal of Theoretical Biology*, 229 (4), 523-537. <https://doi.org/10.1016/j.jtbi.2004.04.037>
- [32] Cox, D. A., Little, J., & O'Shea, D. (2015). *Ideals, varieties, and algorithms*. Springer. doi:[10.1007/978-3-319-16721-3](https://doi.org/10.1007/978-3-319-16721-3).
- [33] Dolzmann, A., & Sturm, T. (1997). Redlog: Computer algebra meets computer logic, *ACM SIGSAM Bulletin*, 31 (2), 2-9. <https://doi.org/10.1145/261320.261324>
- [34] Montes, A. (1998). Algebraic solution of the load-flow problem for a 4-node electrical network, *Mathematics and Computers in Simulation*, 45 (1-2), 163-174. [https://doi.org/10.1016/S0378-4754\(97\)00092-X](https://doi.org/10.1016/S0378-4754(97)00092-X)
- [35] Sit, W. Y. (1992). An algorithm for solving parametric linear systems, *Journal of Symbolic Computation*, 13 (4), 353-394.
- [36] Kalkbrenner, M. (1997). On the stability of Gröbner bases under specializations, *J. Symb. Comput*, 24(1), 51-58. <https://doi.org/10.1145/3666000.3669718>