Constraint satisfaction and global optimization in robotics

Arnold Neumaier

Universität Wien

and

Jean-Pierre Merlet

INRIA Sophia Antipolis

The design, validation, and use of robots poses a number of challenging

- **global constraint satisfaction and**
- **global optimization problems**
- **in dimensions ranging from a few to several hundreds,**
- **with quadratic, polynomial, or transcendental constraints.**

The talk will discuss background, formulation, and solution for some of these problems.

This work is part of the COCONUT project

a European Union initiative

for the integration of different

global optimization techniques.

www.mat.univie.ac.at/coconut

Then God said, "And now we will make human beings; they will be like us and resemble us."

Genesis 1:26.

"They speak one language; this is just the beginning of what they are going to do. Soon they will be able to do anything they want!"

Genesis 11:6.

A robot is a machine designed to perform

repetitive (but more and more frequently also intelligent) tasks that not long ago were the domain of human activities only,

and also tasks that are beyond the limitations of human working conditions.

Robotics is the science of designing, building, analyzing and controlling robots; it deals with all aspects of this.

Computational robotics problems may be classified into the following, partially overlapping aspects:

- Architecture
- Geometry
- Geometric kinematics
- Differential kinematics
- Statics
- Dynamics
- Trajectory planning
- Sensing
- Control
- Performances evaluation
- Design
- Calibration

- **Architecture** is concerned with defining the topology, the basic mechanical connectivity pattern of a robotic system.

- **Geometry** treats questions such as the numerical dimensions of the robot parts, limits on lengths and angles, intereference between legs, or the accessible workspace.

- Geometric kinematics is about calculating the state of a robot from measurements (direct kinematics) or poses (inverse kinematics), and associated questions of accuracy and singularities.

- Differential kinematics is the study of changes of geometry and the resulting velocities and angular velocities, accelerations and angular accelerations, as far as dynamical questions can be ignored.

- Statics treats the relations between forces and the geometry, including stiffness considerations.

- Dynamics is concerned with the behavior of a robot in motion, and the dynamical response of the robot to loads and other forces.

- <u>Trajectory planning</u> treats the problem of finding suitable trajectories between points in the workspace, and analyzing the behavior of the robot along such a trajectory.

- <u>Sensing</u> treats the problem of providing the robot with information about the environment and itself, making it capable of doing certain taks without supervision.

- <u>Control</u> deals with issues related to ensuring a desired motion of the robot in the presence of uncertainties in the model parameters and inaccuracies of the models used.

- <u>Performance evaluation</u> is about checking the extent to which user-specified goals are met by a given robot.

- Design is about meeting user-specified goals by choosing the right architecture (topological synthesis) and geometry (dimensional synthesis) for a robot to satisfy well-defined performance criteria.

- Finally, after building a model according to a specified design, calibration is the task of figuring out how closely the actual robot follows the model, and how to correct the design parameters to more closely match those of the real robot.

These topics pose a large number of

- **challenging highly nonlinear algebraic problems**
- **in a moderate number of variables.**
- **usually with multiple solutions that may all be of interest.**
- **Safety considerations require a worst case analysis of the possible scenarios.**

$$\Longrightarrow$$

- **global optimization problems**
- **or constraint satisfaction problems**

The tasks typically arising lead to numerical problems of three different types:

- local problems
- global problems
- bilevel problems

Local problems

Real time on-line activities (~ 1 millisecond) are generally local problems that are easy to solve:

- expression evaluations
- finding a solution of small systems of equations close to a given starting point (continuation)

Some of these problems have, however, close multiple solutions, in which case they belong to the next category, and fast solution techniques are lacking except in simple cases.

Global problems

- Tolerance problems:
  a property must be checked for all points in some higher-dimensional space,
- Constraint satisfaction problems:
  one (existence) or all (covering) points must be found that satisfy a given set of constraints
- global optimization problems:
  point(s) must be found satisfying given constraints and attaining the absolute extreme values (global minimum or global maximum or both) of some variable or objective function.

Global problems (ctd.)

Depending on the complexity of the defining expressions and the dimension and size of the relevant search space:

- such problems may take hours of computation time (much more if the solution set is large or has a complicated boundary)

- in large scale and other challenging cases it may be difficult or impossible to get a complete solution.

Bilevel problems

- usually arise from design questions
- constraints are themselves tolerance problems or global optimization problems
- multiple solutions generally exist, have significantly different geometric and mechanical properties
- some of the solutions may be more suitable to accommodate other, unmodelled design criteria.

- incomplete information from successful local searches is already useful.

Bilevel problems (ctd.)

Design problems often belong to the hardest tasks since they may require repeated solutions of other global problems that determine the performance criteria.

Depending on the complexity of the design goals, their solution may take many computer days.

This talk ignores

- most aspects relating to artificial intelligence (sensing and planning)
- most of the dynamical aspects (dynamics and control).

I only look at modeling and design problems for robots, with emphasis on parallel robots, in particular so-called Gough platforms.

There is a bottomless pit of further challenging problems . . .

Architecture

The mechanical subsystem of a robot consists of

- a system of mechanical links and joints,
- designed such that one or more end-effectors (e.g., grippers, finger tips) can perform tasks of interest.

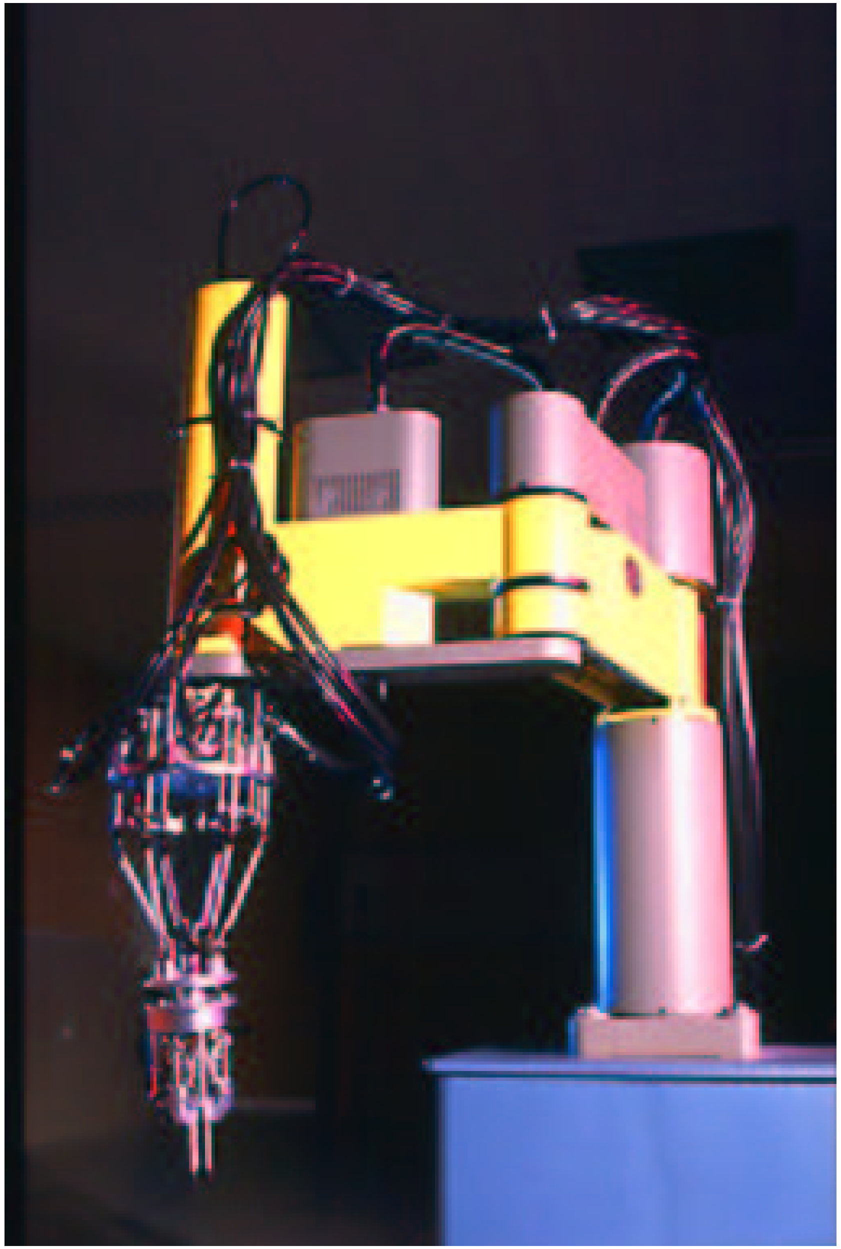
To move the robot, some of the joints – the actuated joints – can be changed by actuators (motors), within certain limits depending on the actual construction of the robots.

Coordinated motion is made possible by feeding measured joint coordinates obtained by various kinds of <u>sensors</u> into a <u>controller</u>.

The controller uses goals related to the task to be done, provided by the information processing system (the robot's mind).

The latter compute and send appropriate orders to the actuators.

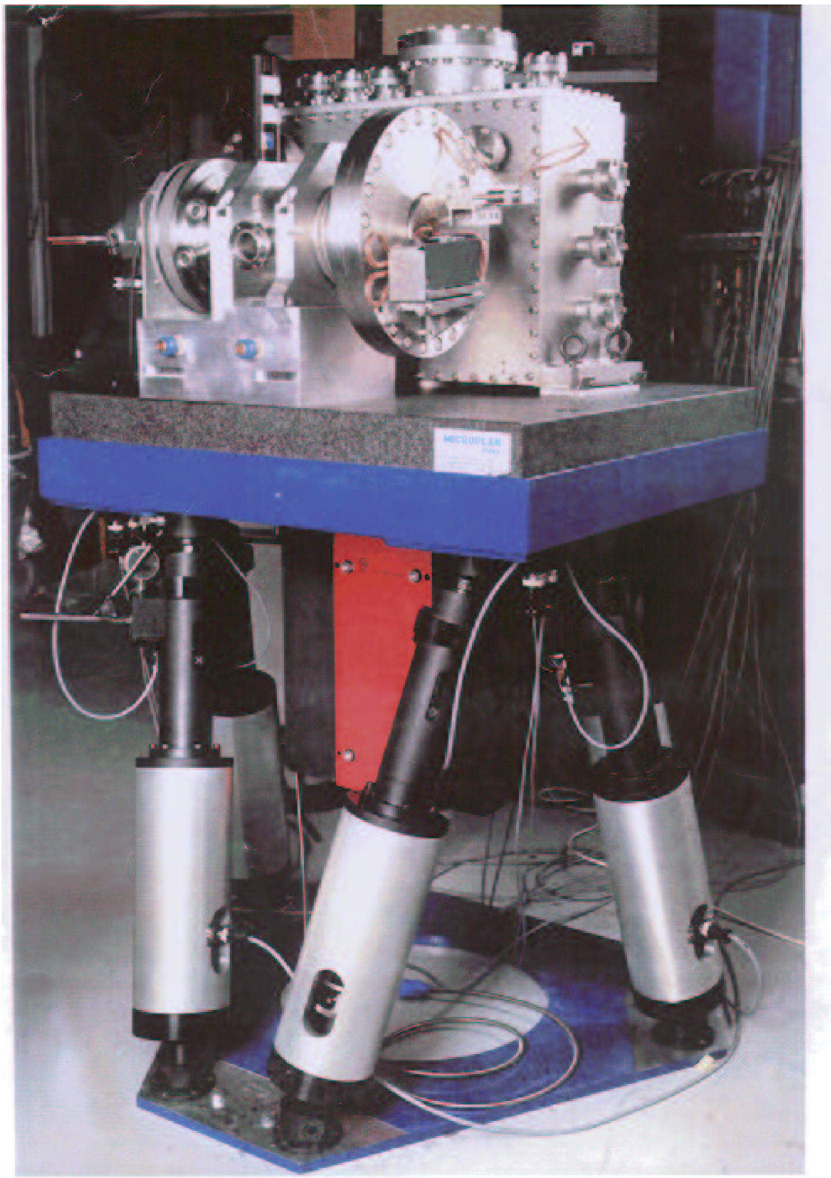
These exert forces and torques to move the joints towards the desired position.



Architecture II

Many different architectures are possible.

A parallel manipulator, added as a wrist to a SCARA serial robot with 4 degrees of freedom (yellow).



A parallel robot at the European Synchrotron Radiation Facility in Grenoble, can manipulate loads of up to 1000kg with an accuracy of better than $1\mu m$.

For example, a <u>Gough platform</u> consists of two rigid bodies, a fixed base and a moving platform connected by six legs of adjustable length to give the pose (position and orientation) of the platform six degrees of freedom.

To determine the current pose, a minimal vector of measured <u>joint coordinates</u> consists of six independent coordinates, e.g., six leg lengths.

Architecture III

For a quantitative treatment, realistic robot models are needed.

In a frequently used idealization, the mechanical subsystem consists of a number of rigid bodies called <u>links</u>, coupled by so-called <u>kinematic pairs</u> = <u>joints</u>.

- <u>prismatic pairs</u> (P) are links joined by a sliding prism with a definite direction,
- <u>revolute joints</u> (R) are links joined by a joint with a definite rotation axis.

If the kinematic pair can be moved by a motor it is called a linear (P) or rotary (R) actuator; other kinematic pairs perform only passive motions and are called passive; they adjust themselves to the forces imposed by the loads and actuators.

Architecture IV

To describe an open kinematic chain, *Denavit & Hartenberg* **introduced on each link a special coordinate system.**

A point with coordinates $x_i \in \mathbb{R}^3$ in the coordinate system of link $i > 0$ (for links $0, \ldots, n$) has coordinates

$$x_{i-1} = Q_i x_i + c_i \quad (i - 1, \ldots, n)$$

with rotation matrices $Q_i \in SO(3)$ and translation vectors $c_i \in \mathbb{R}^3$ of special structure.

Writing

$$Q_{12}(\theta) := \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad Q_{23}(\alpha) := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix},$$

we have

$$Q_i = Q_{23}(\alpha_i), \qquad c_i = \begin{pmatrix} a_i \\ 0 \\ \theta_i \end{pmatrix}$$

for a prismatic pair in linear position θ_i,

$$Q_i = Q_{12}(\theta_i)Q_{23}(\alpha_i), \qquad c_i = Q_{12}(\theta_i) \begin{pmatrix} a_i \\ 0 \\ b_i \end{pmatrix}$$

for a revolute joint in angle position θ_i.

Here α_i, a_i and (for revolute joints) b_i are fixed numbers describing the geometry of the kinematic pair, while the joint coordinate θ_i describes the degree of freedom in positioning the pair.

The coordinate transformations

$$x_{i-1} = Q_i x_i + c_i \quad (i - 1, \ldots, n)$$

completely determine the relative position of the final link (usually the end-effector) with respect to the initial link (usually the base).

Architecture V

A separable parallel robot has 6 legs in which the pose (c, Q) of the end-effector Q is related to the actuated joint coordinates θ_i by six scalar equations

$$H_i(d_i^c, \theta_i, c + Q d_i^p - d_i^b) = 0 \quad (i = 1, \ldots, 6).$$

Here d_i^b and d_i^p are the local coordinates of the endpoints of leg i on the base and the platform, respectively, and d_i^c is a vector of coefficients determining the details of H_i.

Together with the six equations

$$(Q^TQ)_{ik} = \delta_{ik} = \begin{cases} 1 & \textbf{if} \quad i = k\textbf{,} \\ 0 & \textbf{otherwise} \end{cases}$$

and the orientation constraint

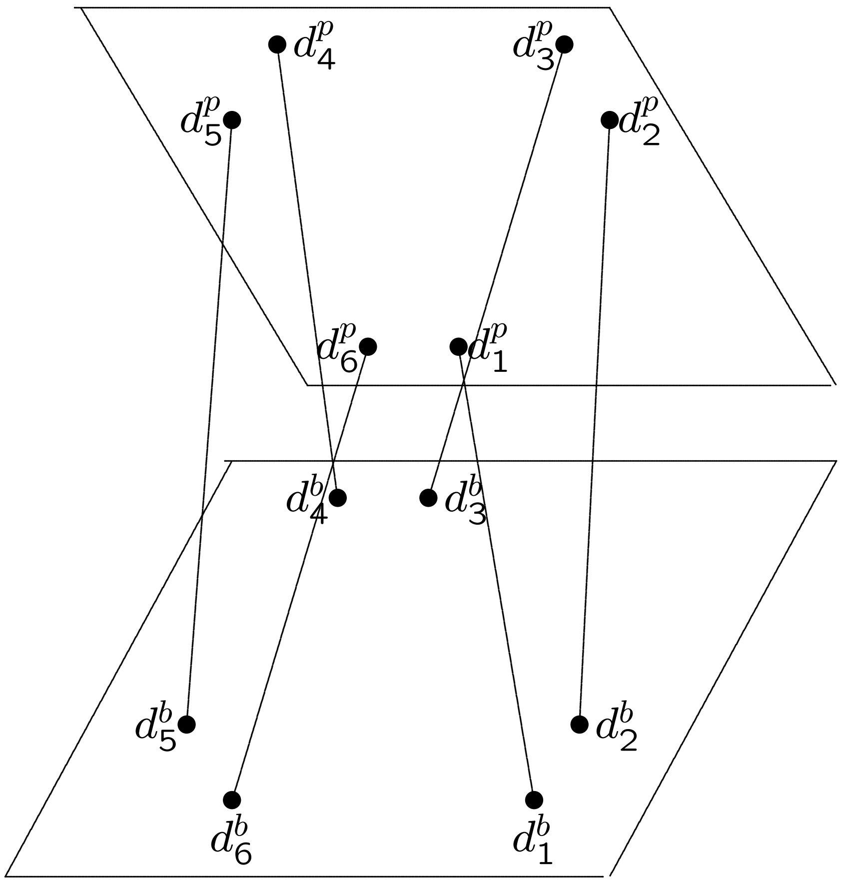
$$\det Q > 0,$$

the separable parallel robot equations

$$H_i(d_i^c, \theta_i, c + Qd_i^p - d_i^b) = 0 \qquad (i = 1, \ldots, 6).$$

give a complete description of a practically relevant class of model parallel robots.

platform
= end-effector



base

A <u>Gough platform</u> is characterized more specifically by the relations

$$\|c + Qd_i^p - d_i^b\|^2 - (l_i + \theta_i)^2 = 0,$$

corresponding to a platform with 6 ideal legs (of type RR<u>P</u>S).

For an ideal leg, the actuated pair is prismatic and the other pairs are revolute, with DH parameters given by

i	a_i	b_i	α	**type**
1	a_1	b_1	α_1	R
2	**0**	**0**	α_2	R
3	**0**	θ_3	**0**	P
4	**0**	b_4	α_4	R
5	**0**	**0**	α_5	R
6	a_6	b_6	α_6	R

Similarly, separable parallel robots with legs of type PRRS (an active wrist or Hexaglide) satisfy equations of the form

$$\|c + Qd_i^p - d_i^b - \theta_i u\|^2 - l_i^2 = 0,$$

and separable parallel robots with legs of type RRRS (a Hexa robot) satisfy equations of the form

$$\|c + Qd_i^p - d_i^b - (\cos\theta_i)u - (\sin\theta_i)v\|^2 - l_i^2 = 0.$$

To design, analyze, and operate a robot, a large number of tasks must be solved. In the following, we consider idealized versions of some of the computational tasks involved. We begin with the problems directly related to the geometry of the robot.

Geometry

- The generalized coordinates s code the pose of the end-effector.
- The equation of state

$$F(s, \theta) = 0$$

relates the pose to the joint configuration θ. For a Gough platform, it has the explicit form $\theta = F(s)$.
- The measurement equation

$$y = P\theta,$$

relates θ to the measurement vector y containing the values obtained by internal sensor measurements.

The equation of state contains all information about the geometry of measured joints and the end-effector. . .

. . . apart from limits of the ranges on joint variables, global size constraints for the robot, and inequality constraints ensuring the absense of interference between different links.

The latter usually need more detailed information about the way the robot is built.

Note that although of immense practical usefulness, all models discussed here are idealizations that neglect certain aspects which must be taken care of in practice, usually by adjusting the basic analysis obtained from the idealized model.

Generalized coordinates

The generalized coordinates code the pose of the end-effector. Their choice is a matter of convenience.

Quaternion parameterization: $s = \binom{r}{c}$

- c is the position of a specific point of the end-effector relative to the base.
- r encodes the rotation in terms of $r \in \mathbb{R}^3$ with real $r_0 = \sqrt{e - r^T r}$:

$$Q = Q_e[r] = I + \frac{2}{e} \begin{pmatrix} -r_2^2 - r_3^2 & r_1 r_2 - r_0 r_3 & r_1 r_3 + r_0 r_2 \\ r_1 r_2 + r_0 r_3 & -r_1^2 - r_3^2 & r_2 r_3 - r_0 r_1 \\ r_1 r_3 - r_0 r_2 & r_2 r_3 + r_0 r_1 & -r_1^2 - r_2^2 \end{pmatrix}$$

Workspace

The mechanical architecture of the robot or technological constraints such as limits on the joint motions impose that the end-effector cannot reach any pose.

Simple bounds on the joint variables θ_i, and perhaps bounds the angles between legs and platforms, chosen such that no leg interference is possible within these limits, are taken into account in the modeling process.

The constraints are defined by a vector of constraint inequalities

$$C(s, \theta) \in \mathbf{C},$$

where C is a box, defining the state space $\mathcal{S}$ of the robot.

- **6 dimensions or less**
- **curved because of rotations**

Workspace II

6D workspace = projection

$$\mathcal{W} = \{s \mid (s, \theta) \in \mathcal{S}\}$$

of the state space to the space of possible poses.

It contains all platform poses realizable by the actuators within the given constraints.

Constraints on the position c are generally simple bounds resulting from the need that the robot has to work in an environment of given size.

Constraints on rotations are more varied.

$$r^2 \leq \frac{e}{2}(1 - \cos\alpha)$$

limits the rotation to an angle $|\theta| \leq \alpha$ around its axis.

$$r_1^2 + r_2^2 \leq r_3^2 \tan^2\alpha$$

limits the rotation axis to a circular cone with half opening angle α around the z-axis.

$$r_1^2 \tan^2\alpha \leq r_2^2 + r_3^2$$

keeps the rotation axis away from a circular cone with half opening angle α around the x-axis.

Workspace III

Because of the limited visualizability of six dimensions, engineers want to see certain projections of the 6D workspace.

Typically, three of the pose parameters will have a fixed value or they must lie within given ranges.

- **<u>Constant orientation workspace:</u> For a given orientation r, find the set of positions c such that $\binom{r}{c} \in \mathcal{W}$.**

- **<u>Orientation workspace:</u> For a given position c, determine the set of orientations r such that $\binom{r}{c} \in \mathcal{W}$.**

- **Total orientation workspace:** For a given set r of desired orientations, find the set of positions c such that $\binom{r}{c} \in \mathcal{W}$ for all $r \in \mathrm{r}$.

- The **dextrous workspace** is the important special case in which r is unrestricted beyond its definition.

- **Maximal workspace:** Find the set of positions c for which there exists at least one orientation r such that $\binom{r}{c} \in \mathcal{W}$.

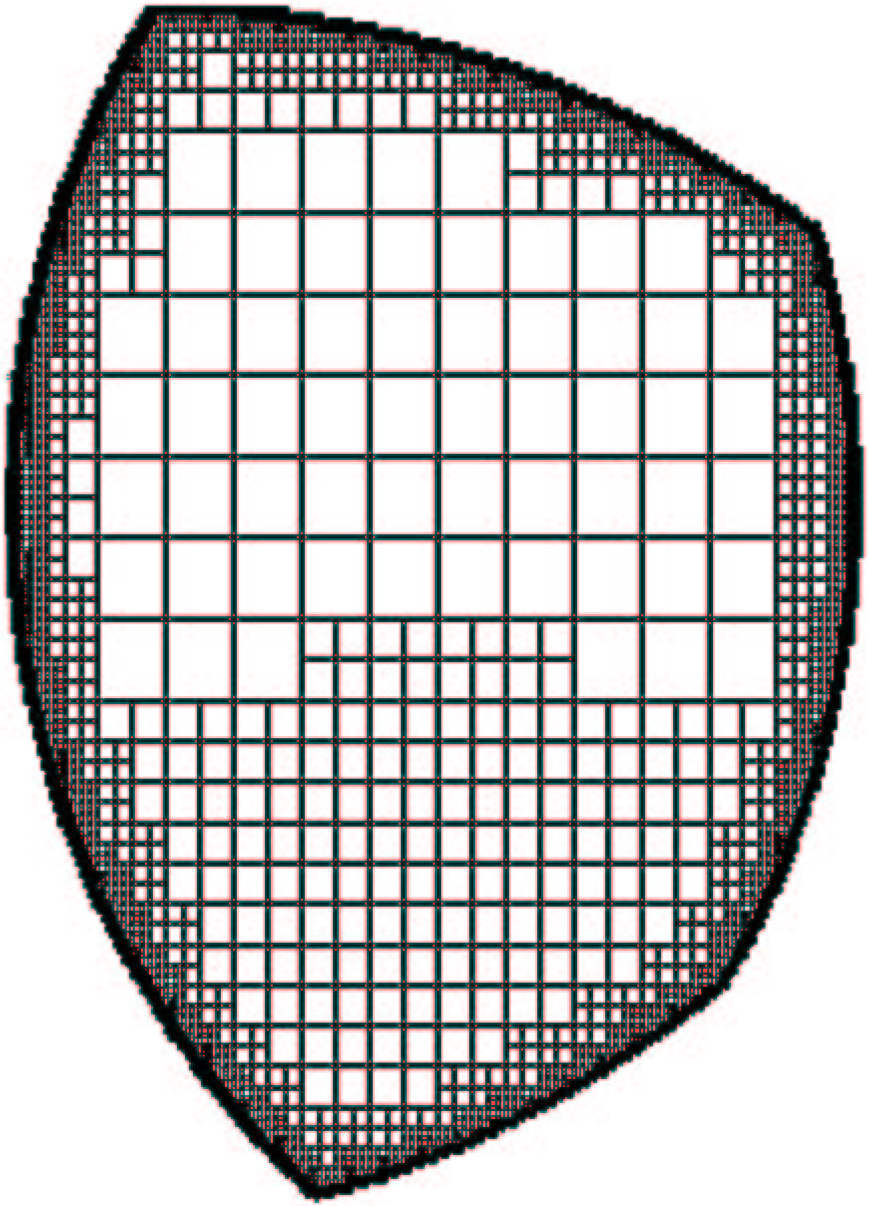
Workspace IV

These tasks are constraint satisfaction problems of the covering type.

If only limits on the actuated joints are considered, the constant orientation workspace for a separable parallel robot is an intersection of six circular rings and may be computed in a few milliseconds.

The task becomes more complex once others types of constraints are taken into account (passive joints motion limits, leg interference).

The most difficult task is the calculation of the maximal workspace that may take hours of computation time for manipulators with large ranges for the joint variables.



A horizontal 2D cross section through the maximal (6D) workspace of a Gough platform

An associated problem is to define a geometrical object such as a cube or a sphere and to determine what is the largest such object enclosed in the workspace of the robot. This can be solved once the covering has been obtained, but how to do it efficiently has not been investigated.

Workspace V

Verifying the reachability of a planned workspace is the task to decide whether a specific planned work space $\mathcal{W}_0$ is contained in the full 6D workspace $\mathcal{W}$.

This can be solved in two steps:

- **First verify the existence of some point $x_0 \in \mathcal{W}_0$. One can frequently find such a point by local methods; only in hard cases a global search is needed.**

- **Then show that**

$$F(s,\theta) = 0,\ \ s \in \mathcal{W}_0,\ \ C(s,\theta) \in \partial\mathrm{C},$$

has no solution. This <u>boundary–freeness</u> condition is a global constraint satisfaction problem and therefore hard.

If both parts are successfully resolved, continuity implies reachability.

Geometric kinematics

Geometric kinematics is concerned with

- solving the equation of state for the states (s, θ), given partial information

- assessing the accuracy with which the stated can be determined from noisy measurements.

Direct and inverse kinematics

- The <u>direct kinematics</u> problem is the task

- to find the set of possible states (s, θ) for given measurements $y = P\theta$ of the sensed part $P\theta$ of the joint configuration θ, and

- to determine which of these corresponds to the actual pose of the robot.

The goal of the <u>inverse kinematics</u> problem is

- to determine the set of possible states (s, θ) for given generalized coordinates s, and

- to determine which of these corresponds to the actual joint configurations of the robot.

For a robot in which all modelled joint variables are sensed:

- **direct kinematics means to find s from θ,**

- **inverse kinematics means to find θ from s.**

For serial robots, direct kinematics is straightforward, solved by the evaluation of a matrix product. the pose is uniquely determined by the joint configuration.

The inverse kinematics of serial robots involves 6 non-linear equations in 6 variables, and may be difficult.

For example, the inverse kinematics of the general 6R robot arm (with 6 revolute joints) has up to 16 solutions; finding them all is a global problem.

In contrast, the inverse kinematics of separable parallel robots is simple, amounting to the solution of univariate equations. In particular, the inverse kinematics of a Gough platform is trivial, since

$$\|c + Qd_i^p - d_i^b\|^2 - (l_i + \theta_i)^2 = 0$$

gives the unique joint configurations for given generalized coordinates.

On the other hand, the direct kinematics of parallel robots involves 6 nonlinear equations in 6 variables, and is usually difficult. For example, the direct kinematics of a general Gough platform may have up to 40 real solutions.

The currently known algebraic methods for finding them all are numerically unstable and need slow multi-precision arithmetic to produce reliable results.

The direct (inverse) kinematics problem may be local if we have information on the possible locations of the end-effector (or the joints).

This is the case in <u>real-time control</u> (continuation).

In the global problem we have no information on the current pose (wake-up).

This problem has attracted the interest of researchers in recent years, and there has been a lot of progress made in the determination of all solutions of this problem.

The most efficient solving algorithm will currently find all the solutions in a computation time varying between a few seconds and up to 30 minutes depending on the geometry of the robot.

Dexterity

Dexterity is the ability to flexibly and accurately place the end-effector in a given pose.

The flexibility is quantified by the dextrous workspace discussed before.

The accuracy may be quantified by dexterity indices that may be defined in numerous ways.

Although widely used for the design of manipulators, the reliable evaluation of dexterity indices is an open problem.

Monte Carlo studies are still the state of the art.

The pose of a robot, as estimated by direct kinematics, is inaccurate. . .

. . . since it is based on more or less accurate measurements $y \approx P\theta$ of the measured part $y = P\theta$ of the joint configuration θ.

Hence errors on the sensor measurements induce an error $\triangle s$ on the positioning of the robot.

In the customary linear approximation, pose errors Δs and measurement errors $\Delta\theta$ are related by an equation of the form

$$A(s,\theta)\Delta\theta = B(s,\theta)\Delta s,$$

where A, B are matrices with coefficients that depend on the state of the robot.

For a serial robot this relation has the form

$$\Delta s = J(s, \theta)\Delta\theta$$

with a state-dependent matrix J, called the (kinematic) Jacobian matrix of the robot.

(In spite of the name, it is not a true Jacobian, i.e., not the derivative of a vector function.)

For a Gough platform this relation has the form

$$\Delta\theta = J^{-1}(s,\theta)\Delta s.$$

The matrix J^{-1} (labelled this way by analogy) is not necessarily invertible; it is called the <u>inverse Jacobian matrix</u> of the robot.

Tasks

- For the error measures $\varepsilon(\Delta s)$ of interest, find the maximal error over a workspace of specified type given bounds on the sensor accuracy.
- Find the sensor accuracy needed to guarantee a specified maximal error over a given workspace.
- Find the state (s, θ) in which the maximal error is smallest.
- Find the set of states in which Δs lies within a specified region.

Singularities

A singularity is a state (s, θ) where one of the matrices in the sensitivity equations

$$A(s,\theta)\Delta\theta = B(s,\theta)\Delta s$$

is singular.

Singularities within the 6D workspace indicate trouble that must be avoided.

At a singularity of A, a robot loses one degree of freedom. (Cannot happen for a Gough platform since $A = I$.)

At a singularity of B, a robot gains one degree of freedom. (Cannot happen for a serial robot since $B = I$.)

But the Hexa robot, for example, can have both kinds of singularities.

Joint singularities of A and B compound the difficulties created by singularities in A and B.

At a singularity of A, a robot <u>loses</u> one degree of freedom (or more): Since $A\Delta\theta$ is restricted to a subspace, Δs cannot go in some direction no matter how joint coordinates are changed.

If higher order terms are taken into account, one only loses a halfspace of possible pose changes: Small changes ε in pose space require large changes of order $\sqrt{\varepsilon}$ or more in joint space, forcing $\varepsilon > 0$.

For example, if an arm is fully extended, the hand cannot move in the direction of the arm.

Similarly, at a singularity of B, a robot <u>gains</u> one degree of freedom (or more): it can move while the actuators are locked.

As a result, forces may have uncontrolled effects.

If higher order terms are taken into account, one sees that does not quite gain a degree of freedom but is at a branch point in whose neighborhood multiple solutions exist.

Moreover, significant changes of order $\pm\delta$ in pose space are possible with only tiny changes δ^2 in joint space, and huge joint forces may be needed to balance moderate external forces.

Sometimes this has drastic consequences . . .

Conclusion:

In a region of planned usage, e.g., near a given trajectory, or everywhere in the workspace, it is essential to verify the absence of singularities!

This leads to constraint satisfaction problems involving constraints of the form

$$\det A(s, \theta) = 0.$$

Naive use leads to computational nightmares...

...long, symbolic expressions prone to excessive over-estimations.

Singularities II

This problem of analyzing the workspace singularities has been intensively studied recently.

The main problem is that although an analytical form of the matrices is known the expansion of their determinant, although possible, leads to a huge expression.

Several related problems are of interest.

Singularities III

- determine simpler singular conditions: this has been obtained through Grassmann geometry

- determine what will be the infinitesimal motion at a given singularity

- determine the presence of singularities within a given workspace

- **verify that a given trajectory is singularity free**

- **define a geometrical object such as a cube or a sphere and determine what is the largest such object that is enclosed in the workspace and is singularity free**

Design

All problems so far have been considered for a robot having a given geometry.

This geometry is defined by a vector d of parameters (e.g., the location of the attachment points of the legs for a Gough platform).

The numbers of design parameters may be relatively large (up to 138 for a Gough platform).

Performance evaluation

Performance indices are used either to evaluate the performance of a fixed robot or to compare the performance of different robots according to

- geometric criteria
- velocity criteria
- statics criteria
- dynamics criteria

A <u>performance index</u> is a real number I defined by an equation

$$H(d, s, \theta, I) = 0,$$

where d is a vector of <u>design parameters</u>, which may

- **be a fixed design**
- **vary over a set of alternative designs**
- **vary over a tolerance margin**
- **vary over a set of alternative designs with tolerance margin**

Performance evaluation II

Evaluation of a performance index for a given design means that we must evaluate:

- the minimal and maximal values of I for s belonging to the workspace of the robot
- the average value of I for s belonging to the workspace of the robot

The workspace in this calculation may be either a user-defined workspace or the maximal 6D workspace.

For comparative performance evaluation, it may not be necessary to calculate exactly the performance index; calculations can be stopped as soon as the ranking of the designs is determined.

Geometric criteria

- overall size
- workspace (volume, shape)
- motion range for the actuators and the passive joints
- maximal positioning errors of the end-effector
- isotropy of the performances
- quality of trajectory following
- transmission factor
- singularities within the workspace

Velocity criteria

- maximal velocities of the actuators and platform
- minimal guaranteed velocity for the moving platform for bounded joint velocities over a given workspace

Statics criteria

- maximal forces or torques on the links
- stiffness
- maximal forces or torques applied to the platform
- energy efficiency
- position of the center of mass

Dynamics criteria

- maximal accelerations of the actuators and platform
- minimal guaranteed acceleration for the platform for limited joint acceleration and velocity over a given workspace

Optimal design

The optimal design problem is to determine the values of the design parameters such that the associated robot presents the best performances for a given task.

Key issue: Parallel robots exhibit potentially high performance in many respects.

But these performances are very sensitive to the geometry of the robot.

E.g., changing the size of the platform by 10 % sometimes changes the value of the minimal stiffness of the robot by 700 %!

Hence it is necessary to perform a careful dimensional synthesis to obtain the desired performances.

The task is therefore to determine the possible designs such that constraints on several performance index are satisfied.

Not all the performance indices that play a role in the design have the same importance.

Usually a compromise must be made according to considerations, some of which may be unknown to the designer.

Thus the designer has to provide a set of possible solutions, with their associated performances.

Ideally one would want to obtain all sufficiently distinct acceptable designs (or at least a good approximation of them).

Robust design

Simplest form: Find d to guarantee $s^l \in W$ for L specified poses $s^l (l = 1 : L)$.

This gives a system of equations

$$F(d, s^l, \theta) = 0 \quad \textbf{for } l = 1 : L$$

that must be solved together with suitable constraints on d and θ.

The problem appears in three versions.

(a) Is there a solution? Find d_0 satisfying

$$F(d_0, \theta) = 0, \quad \textbf{where } F(d, \theta) = \begin{pmatrix} F(d, s^1, \theta) \\ \vdots \\ F(d, s^L, \theta) \end{pmatrix}.$$

This is a (global) constraint satisfaction problem (unless a good initial design is known).

(b) Is there a tolerant solution, allowing the implementation with prescribed accuracy δ? This requires to check that

$$F(d, \theta) = 0, \qquad \|d - d_0\| \le \delta$$

has no solution. Again, this is a (global) constraint satisfaction problem.

(c) Find a maximally tolerant solution. This can be formulated as

$$\begin{aligned} &\min \delta \\ &\text{s.t. } F(d,\theta) = 0, \|d - d_0\| \leq \delta \end{aligned}$$

An arm design problem

A <u>3R manipulator</u> is an open kinematic chain with 4 **links** $0, \ldots, 3$ **and** 3 **revolute joints. It has** 3 **degrees of freedom and** 9 **design parameters. We consider the design requirement that** m **poses with homogeneous representation**

$$T_l^{\mathsf{des}} = \begin{pmatrix} Q_l & c_l \\ 0 & 1 \end{pmatrix} \qquad (l = 1, \ldots, m)$$

(and without loss of generality $T_l^{\mathsf{des}} = 1$ **for some** l**) can be realized by an end-effector mounted on a** $3R$ **manipulator.**

The joints of a 3R manipulator have associated transformation matrices

$$T_i(\theta) = T(\theta, \alpha_i, a_i, b_i) = \begin{pmatrix} Q_i(\theta) & c_i(\theta) \\ 0 & 1 \end{pmatrix},$$

($i = 1, 2, 3$), where

$$Q_i(\theta) = Q_{12}(\theta)Q_{23}(\alpha_i), \qquad c_i(\theta) = Q_{12}(\theta)\begin{pmatrix} a_i \\ 0 \\ b_i \end{pmatrix}.$$

The pose of the end-effector in coordinates relative to link 3 is specified by the transformation matrix T. Then the realizability of the poses can be modeled by the equations

$$T_1(\theta_{1l})T_2(\theta_{2l})T_3(\theta_{3l})T = T_l^{\text{des}} \quad (l = 1, \ldots, m)$$

for suitable actuator positions $\theta_{il}(i = 1, 2, 3; l = 1, \ldots, m)$.

Because of the structure of the transformation matrices, an equation between them represents 6 independent scalar equations for the 6 degrees of freedom of the relative pose.

Thus we have $6m$ equations for $3m$ actuator positions, $3 \cdot 3$ design parameters, and 6 degrees of freedom in T; a total of $3m + 15$ variables.

This lets one expect that it might be possible to construct a $3R$ manipulator realizing $m = 5$ specified poses.

In this case, we have a system of 30 **equations for** 30 **variables. If we introduce variables**

$$s_{il} = \sin\theta_{il},\ \ c_{il} = \cos\theta_{il},\ \ \lambda_i = \sin\alpha_i,\ \ \mu_i = \cos\alpha_i$$

and corresponding constraints

$$s_{il}^2 + c_{il}^2 = 1, \qquad \lambda_i^2 + \mu_i^2 = 1$$

to get rid of the trigonometric terms, we get 18 **additional equations and variables.**

Finally, using a quaternion parameterization for the three rotational degrees of freedom in T introduces another variable r_0 satisfying the constraint

$$r_0^2 + r^2 = 1, \qquad r_0 \geq 0.$$

Thus we end up with 49 polynomial equations for 49 variables. The degree can be reduced to 4 by rewriting it as

$$T(T_l^{\mathrm{des}})^{-1} T_1(\theta_{1l}) = T_3(\theta_{3l})^{-1} T_2(\theta_{2l})^{-1}.$$

Instead of increasing the number of variables to simplify the problem one may also use symbolic computations to eliminate some of the 30 original variables, ending up with a system of 11 highly complicated equations in 11 unknowns.

Another reformulation as a system of quadratic equations gives a total of $3{\cdot}(8m{+}4){+}12m = 36m{+}12$ **variables related by** $12m{-}12$ **linear and** $3{\cdot}(6m{+}m{+}1){+}6m{+}12m = 39m + 3$ **quadratic equations.**

For $m = 5$ **this gives a consistent system of** 48 **linear and** 198 **quadratic equations for** 192 **variables.**

The linear equations can be solved explicitly, resulting in a consistent system of 198 **less sparse quadratic equations for** 144 **variables.**

Artificial limbs

The arm design problem has been solved recently for a particular set of 5 poses needed for useful artificial limbs for people who lost their arm.

- **ALIAS constraint solving environment (J.-P. Merlet)**
- **interval analysis based, modular**
- **constraint propagation**
- **Krawczyk, Kantorovich, linear programming etc.**
- **public domain**
- **+ symbolic methods (Maple, Groebner bases)**
- **+ intelligent switching between different problem representations**

Solution

- 6 geometrically different solutions were found.
- The complete search on 30 dedicated modern PCs took over 300 computer days; real time about 3 weeks.
- checkpointing
- supervised box management
- algorithmic improvements during runtime

Outlook

Parallel robotics is an emerging technology that – like interval analysis – is at a threshold of becoming accepted as a widely used engineering technology.

Because of the much stronger nonlinearities compared to traditional serial robots, it is still regarded with reservation by the robotics industry.

But because of its much better load-weight ratios, it will be the robot technology of the future.

Fast acceptance depends on being able to compute reliable and informative information on the global behavior of parallel robots.

Interval analysis will play a key part in creating this ability.