# CONSTRAINED SYSTEMS IN MECHANICS AND CONTROL AS THEY APPEAR IN ENGINEERING

Elżbieta Jarzębowska
**Institute of Aircraft Engineering and Applied Mechanics**

Warsaw University of Technology
00-665 Warsaw, Nowowiejska 24,
e-mail:elajarz@meil.pw.edu.pl

**PRESENTATION OUTLINE:**

Introduction

1. Constraints in mechanics and modeling of mechanical systems.
1.1. Constraint concept, sources and classifications in mechanics
1.2. Modeling mechanical systems.

2. Non-material constraint sources.

3. Constraints in control theory.
3.1. Constraint concept, sources and classification.

4. A unified constraint formulation.

5. Transition from the constrained system to the control system.
5.1. Kinematic control models.
5.2. Dynamic control models.

6. Examples.

7. Computation problems.

8. Experiments in nonholonomicity.

9. Conclusions.

# INTRODUCTION

**Motivation:**
Constrained (nonholonomic) mechanical systems have been studied in classical mechanics for a long time but they still gather much attention, since numerous applications of nonholonomic systems make the research topic active. There is still a need for a theoretical base for applications. Wheeled vehicles and spacecrafts, or underactuated ground or underwater vehicles, are all nonholonomic systems. Dynamics of nonholonomic systems is the research topic itself but also it is an input to other areas, for example to nonlinear control.

Modeling and control of robotic systems are good examples of areas where material and non-material constraints restrict and/or specify motion and a variety of tasks robots are to perform.

**A little bit of history:**
At the beginning of the XX century it was observed that for some systems, like electro-mechanical systems, motion equations do not have the form of Lagrange's equations. A new trend has been noticed since that time, i.e. the trend of leaving Lagrange's equations approach.
It has been observed that constraints "that can be realized not through a direct contact" could be put on system motions - the earliest formulation of the non-material constraint is attributed to Appell, Mieszczerski and Beghuin.

# CONSTRAINTS IN MECHANICS AND MODELING OF MECHANICAL SYSTEMS

The concept of constraints in classical approach is based on the assumption that constraints are given *a priori* and they are put upon a mechanical system through other bodies or physical systems. They are position and kinematic constraints referred to as material constraints and they are "known" and "given" by Nature.
This understanding of the concept of constraints and their nature is reflected by a common assumption that nonholonomic constraints arise when bodies are in contact with each other and roll without slipping.

One of general classifications divides constraints into unilateral constraints specified by inequalities, and equality constraints specified by equations.
Constraints can be modeled as ideal or non-ideal.

We address ideal equality constraints.

# MATERIAL CONSTRAINTS

**POSITION** constraints are specified by algebraic equations:

$$\varphi_\alpha(t, q_1, \dots, q_n) = 0. \qquad \alpha=1,\dots,a\ ,\ a<n$$

$\varphi_\alpha, \alpha = 1,\dots,a$, are defined on a $(n+1)$-dimensional manifold and have continuous derivatives up to the second order at least.
Usually, material position constraints are of the form $A(q) = 0$.
Position constraints restrict velocities and accelerations of a system.
Position constraints can be eliminated and the dimension of the problem reduced (this is not always reasonable–further applications)

**KINEMATIC** (velocity) constraints are specified by first order differential equations:

$$\varphi_\beta(t, q_1, \dots, q_n, \dot{q}_1, \dots, \dot{q}_n) = 0. \qquad \beta=1,\dots,b,\ b<n$$

$\varphi_\beta, \beta = 1,\dots,b$, are defined on a $(2n+1)$-dimensional manifold and have continuous derivatives. Usually they are presented as $B_1(t, q, \dot{q}) = 0$, where $B_1$ is a $b\times 1$ vector.
Most material kinematic constraints are of the form $B_1(q, \dot{q}) = 0$.

**KINEMATIC** constraints linear in velocities**:**

$$B_1(t, q)\dot{q} + b_1(t, q) = 0,$$

where $B_1(t, q)$ is a $(b\times n)$ matrix and $b_1(t, q)$ is a $b\times 1$ vector.
Kinematic constraints also restrict accelerations.

The classification above is not the only one that exists in the literature. In both mechanics and control there are constrained systems referred to as **Chaplygin** and **non-Chaplygin**.

Nonholonomic constraint equations can be written in the form

$$\dot{q}_i = \sum_{j=1}^{m} \varphi_{ij}(q_1,\ldots,q_n)\dot{q}_j, \qquad i=m+1,\ldots,n,\ m=n-k$$

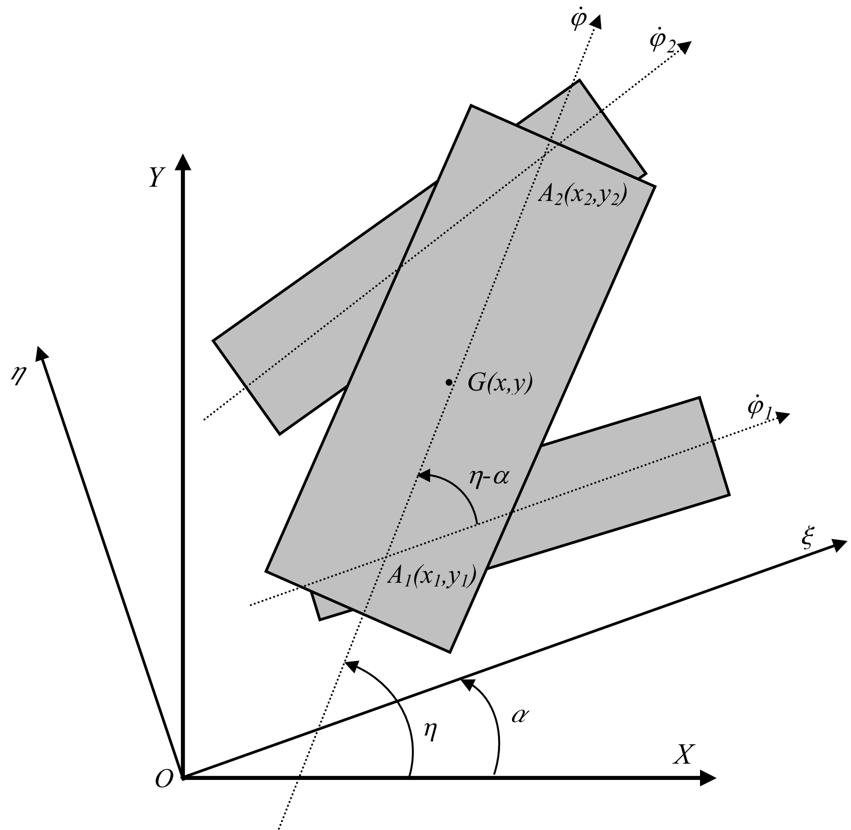
where the first $m$ out of $n$ generalized velocities, in the number equal to the number of system's degrees of freedom, are regarded as **independent**, $(n\text{-}m)$ velocities are **dependent**, and $\varphi_{ij}(q_1,\ldots,q_n)$ are smooth functions of their arguments. Generalized velocities can be partitioned in this way at least locally.

Following the early work of Chaplygin (1897), if constraint functions satisfy certain symmetry properties, namely that they are cyclic in the last $(n\text{-}m)$ generalized coordinates, we obtain Chaplygin nonholonomic constraint equations

$$\dot{q}_i = \sum_{j=1}^{m} \varphi_{ij}(q_1,\ldots,q_m)\dot{q}_j. \qquad i=m+1,\ldots,n$$

Most of nonholonomic systems known in mechanics and control are Chaplygin. However, there is one system, not so well known, the so called Iszlinski example, which is non-Chaplygin

**E. Jarzębowska, N. H. McClamroch, "On Nonlinear Control of the Ishlinsky Problem as an Example of a Nonholonomic Non-Chaplygin System," in *Proc. American Control Conf.*, Chicago, IL, 2000, pp.3249–3253.**




The Ishlinsky example.

The material constraint equations are:

$$\dot{x} = \frac{2az\sin\eta}{w\sin\eta - z\sin(\eta-\alpha)}\dot{\varphi}_1\sin\alpha + R\dot{\varphi}\sin\eta,$$

$$\dot{y} = \frac{2a\sin\eta}{w\sin\eta - z\sin(\eta-\alpha)}[w - z\cos\alpha]\dot{\varphi}_1 - R\dot{\varphi}\cos\eta,$$

$$\dot{\eta} = \frac{2a\sin\eta}{w\sin\eta - z\sin(\eta-\alpha)}\dot{\varphi}_1\sin\alpha,$$

$$\dot{\varphi}_2 = \frac{\sin\eta}{\sin(\eta-\alpha)}\dot{\varphi}_1.$$

The roll rates of the first lower cylinder and of the upper cylinder are control variables: $u_1 = \dot{\varphi}_1, u_2 = \dot{\varphi}$.
It is clear that the kinematic control model is non-Chaplygin

$$\dot{x} = \frac{2az \sin\eta}{w \sin\eta - z \sin(\eta - \alpha)} \sin\alpha u_1 + R \sin\eta u_2,$$

$$\dot{y} = \frac{2a \sin\eta}{w \sin\eta - z \sin(\eta - \alpha)} [w - z \cos\alpha] u_1 - R \cos\eta u_2,$$

$$\dot{\eta} = \frac{2a \sin\eta}{w \sin\eta - z \sin(\eta - \alpha)} \sin\alpha u_1 \ ,$$

$$\dot{\varphi}_1 = u_1,$$

$$\dot{\varphi}_2 = \frac{\sin\eta}{\sin(\eta - \alpha)} u_1,$$

$$\dot{\varphi} = u_2,$$

# NON-MATERIAL CONSTRAINTS THAT COME FROM MODELING

Modeling mechanical systems is goal oriented. Engineers like saying about the “art of modeling”.

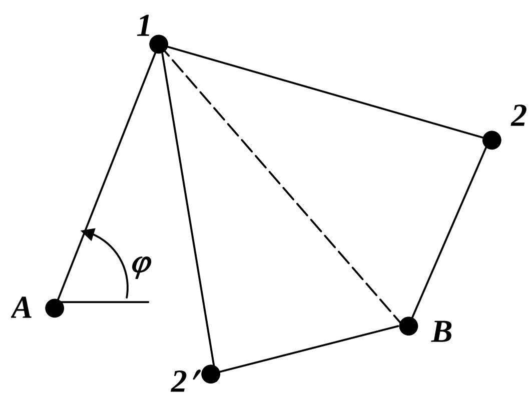
In mechanisms and manipulators modeling, the set of independent coordinates is not always the most suitable. The main problem with the independent coordinates is that they do not describe uniquely the positions of the system elements. Often we prefer an extended set of coordinates whose choice depends on further applications.

There are three major types of coordinates:

Relative coordinates,

Reference point coordinates,

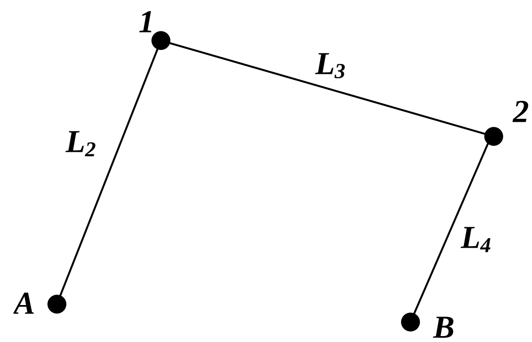
Natural coordinates




One degree of freedom, there are multiple solutions – this generally occurs in multibody systems.

**In natural coordinates:**

They are Cartesian coordinates for points 1 and 2, i.e. $(x_1, y_1, x_2, y_2)$

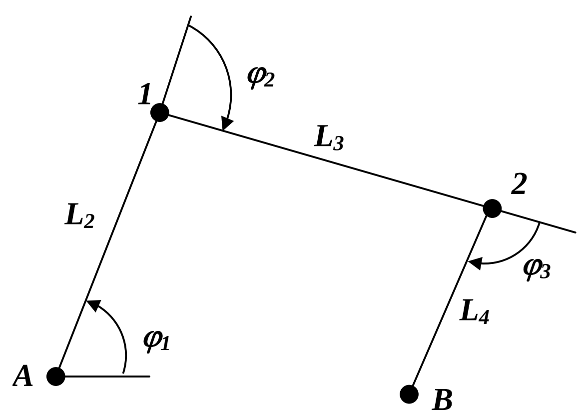



Rigid body condition results in three position constraint equations:

$$(x_1 - x_A)^2 + (y_1 - y_A)^2 - L_2^2 = 0,$$
$$(x_2 - x_1)^2 + (y_2 - y_1)^2 - L_3^2 = 0,$$
$$(x_2 - x_B)^2 + (y_2 - y_B)^2 - L_4^2 = 0.$$

**In relative coordinates:**

Relative coordinates produce no constraints for open chains.

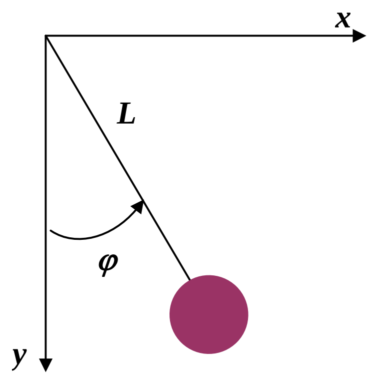



For the closed chain the constraint equations are:

$$L_2 \cos\varphi_1 + L_3 \cos(\varphi_1 + \varphi_2) + L_4 \cos(\varphi_1 + \varphi_2 + \varphi_3) - AB = 0,$$
$$L_2 \sin\varphi_1 + L_3 \sin(\varphi_1 + \varphi_2) + L_4 \sin(\varphi_1 + \varphi_2 + \varphi_3) = 0.$$

**For more examples see J. Garcia de Jalon, E. Bayo, Kinematic and Dynamic Simulation of Multibody Systems, Mechanical Engineering Series, Springer-Verlag, New York, 1994**

Dependent coordinates add additional geometric position constraints. In dynamics, models of systems with geometric constraints are DAE's. What can geometric constraints do?




$$x^2 + y^2 = L^2$$

Motion equations are DAE's of index 3.

In other fields like flight dynamics, there are specialized coordinates that facilitate dynamic modeling and their further applications.

# NON-MATERIAL CONSTRAINT SOURCES IN MECHANICS

♣ The earliest formulation of the non-material constraints was given by Appell (1911). He described them as constraints "that can be realized not through a direct contact". The Appell example:
A particle moves in space *(x,y,z)* and its motion is subjected to a constraint equation

$$\dot{x}^2(t) + \dot{y}^2(t) - \dot{z}^2(t) = \alpha g(t, x, y, z)$$

$\alpha$ is a given constant and $g$ is a known function.

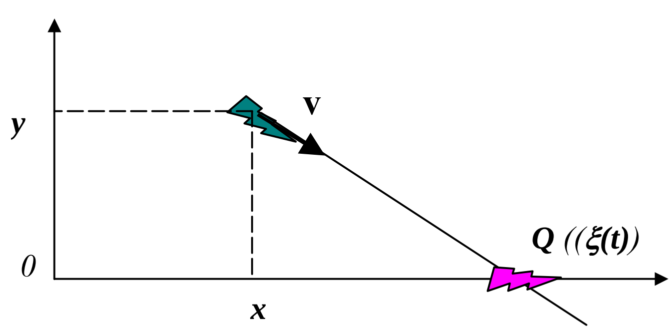
Material constraints are a significant class of motion limitations in engineering practice but there are many problems for which constraints are formulated in a different way. In design or operation problems constraints are formulated before a system is designed. Tasks are such constraints. They are specified first and then we develop dynamic or control models of systems with these constraints.

**Generally, sources of these constraints are not in other bodies and they may arise as performance, design, operation, control or safety requirements. They can be formulated in the form of algebraic or differential equations, or inequalities.**
First ideas were introduced by Mieszczerski at the beginning of the 20-th century. Beghuin developed a concept of **servo-constraints**. These ideas were limited to first order constraint equations. These new "constraint sources" were a motivation to call constraint equations formulations like

$$\varphi_\beta(t, q_1, \ldots, q_\sigma, \dot{q}_1, \ldots, \dot{q}_\sigma) = 0 \qquad \sigma=1,\ldots,n,\ \beta=1,\ldots,b,\ b<n.$$

or $B_1(t, q, \dot{q}) = 0$, where $B_1$ is a $b$-dimensional vector.

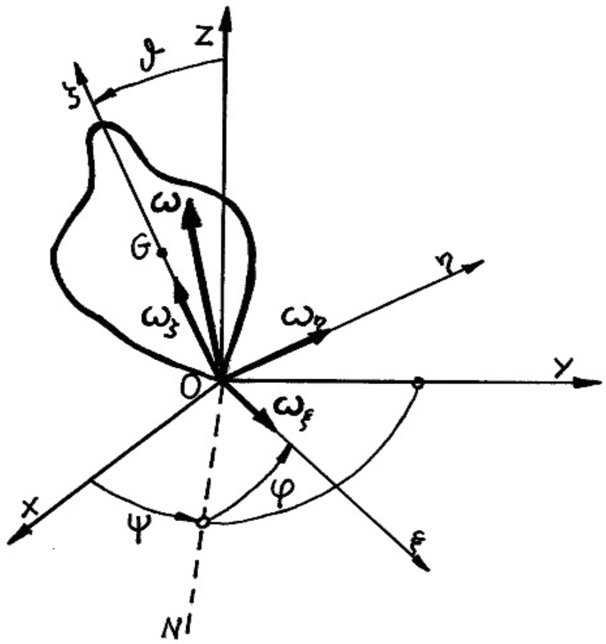
♣ Galliulin and Korenev worked on missile control and guidance (1964). A concept of a program motion appears in the context of a trajectory tracking (tracking a moving target).
Example of the problem of this type is: A target $Q$ moves along $ox$ a prescribed motion $\xi(t)$. A following particle moves in the $(x,y)$ plane in such a way that its velocity vector is directed towards $Q$. The constraint equation for the follower is




$$y\dot{x} + (\xi - x)\dot{y} = 0.$$

The follower will move along a curve of pursuit.

♣ Grioli's example (1972):




Grioli's theorem: necessary and sufficient conditions for a body to perform the pseudoregular precession are:

$$(p\dot{q} - \dot{p}q) + r\left(p^2 + q^2\right) - \lambda\left(p^2 + q^2\right)^{3/2} = 0,$$

where $\lambda = const.,\quad p = \omega_\xi,\quad q = \omega_\eta,\quad r = \omega_\zeta,$

$$\omega_\xi = \dot{\psi}\sin\vartheta\sin\varphi + \dot{\vartheta}\cos\varphi, \qquad \omega_\eta = \dot{\psi}\sin\vartheta\cos\varphi - \dot{\vartheta}\sin\varphi,$$

$$\omega_\zeta = \dot{\psi}\cos\vartheta + \dot{\varphi},$$

and $\varphi, \psi, \vartheta$ are Euler angles. Inserting ω's into the Grioli condition we obtain nonholonomic constraint equations of the second order

$$\ddot{\psi}\dot{\vartheta}\sin\vartheta - \ddot{\vartheta}\dot{\psi}\sin\vartheta + 2\dot{\psi}\dot{\vartheta}^2\cos\vartheta + \dot{\psi}^3\sin^2\vartheta\cos\vartheta$$
$$-\lambda\left(\dot{\psi}^2\sin\vartheta + \dot{\vartheta}^2\right)^{3/2} = 0.$$

♣ Hogan's example (1984) - the third time derivative of a limb coordinate influences its motion smoothness

# CONSTRAINTS IN CONTROL THEORY

1. Material constraints – rolling without slipping – wheeled vehicles, multifinger hands.
2. Conservation laws – the angular momentum conservation for free floating space manipulators.
3. Tasks (performance goals) – robots and manipulators do work and it may be described by tasks specified by the constraint equations (trajectory).
4. Design or control constraints - manipulators and robots with underactuated degrees of freedom.
5. Other design, control, operation constraints for robots and manipulators:
   - in navigation of wheeled mobile robots, to avoid the wheel slippage and mechanical shock during motion, dynamic constraints such as acceleration limits have to be imposed,
   - in path planning problems, for car-like robots, to secure motion smoothness two additional constraints are added: they are put on a trajectory curvature and its time derivative so additional constraints of the second and third order are imposed,
   - in manipulator trajectory tracking, jerk must be limited for reducing manipulator wear, improving tracking accuracy, for smoothed actuator load, for reducing the excitation of the resonant frequencies. Low jerk trajectories can be tracked faster and more accurately,
   - bounded linear velocity,
   - bounded angular velocity,
   - constraints on control inputs,

- bounded lateral acceleration – e.g. path tracking experiments depend on the precision of the odometry. If the lateral acceleration of the vehicle is too large, the wheels can loose close contact to the ground and the odometry data is no longer meaningful,
- others …..

6. Programmed constraint – a concept of a non-material constraint suitable to specify tasks and motion limitations by equations (Jarzębowska).
   - tracking a trajectory (a non-material constraint never incorporated into the system constrained dynamics),
   - robot motion with prescribed motion characteristics,
   - prescribed end-effector motion and many other TASKS.

# A UNIFIED CONSTRAINT FORMULATION

Other examples of non-material constraints appear in development of mechanics models for control applications.

The overview of constraint classifications in mechanics and a variety of requirements on system's motions reported in the literature can be summarized as follows:

1. Many problems are formulated as synthesis problems and motion requirements may be viewed as constraints put on a system before it is designed and put into operation. These constraints can be non-material.
2. Constraints that specify motion requirements may be of orders higher than one or two.
3. Non-material constraints may arise in modeling and analysis of electromechanical and biomechanical systems.
4. No unified approach to the specification of non-material constraints or any other unified constraint has been formulated.

These conclusions were a motivation to present an extended concept of a constraint.

**Definition 1:** A programmed constraint is any requirement put on a physical system motion specified by an equation.

**Definition 2:** A programmed motion is a system motion that satisfies a programmed constraint.

Based on definitions 1 and 2 we classify the programmed constraints as follows:

1. Position programmed constraints:

$$f_{\alpha}(t, q_1, \ldots, q_n) = 0, \qquad \alpha = 1, \ldots, a, \; a < n$$

or $A_1(t, q) = 0$, $A_1$ is an $a$-dimensional vector. The programmed constraints also restrict allowable velocities and accelerations.

2. Kinematic programmed constraints:

$$f_{\beta}(t, q_1, \ldots, q_n, \dot{q}_1, \ldots, \dot{q}_n) = 0, \qquad \beta = 1, \ldots, k, \; k < n$$

or $B_1(t, q, \dot{q}) = 0$, where $B_1$ is a $k$-dimensional vector.

Mathematical relations for the programmed constraints are the same as those for material constraints but their interpretation is absolutely different. The programmed constraints are put upon a system in order to specify its motion.

3. High order programmed constraints:

$$G_\beta(t, q_1, ..., q_n, \dot{q}_1, ..., \dot{q}_n, ..., q_1^{(p)}, ..., q_n^{(p)}) = 0, \qquad (*)$$

$$\beta = 1, ..., k, \ k < n$$

which we write as

$$B_\beta(t, q, \dot{q}, ..., q^{(p)}) = 0,$$

where $p$ is a constraint order and $B_\beta$ is a $k$-dimensional vector. Equations (*) can be nonlinear in $q^{(p)}$.

Differentiation of (*) with respect to time, until the highest derivative of a coordinate is linear, results in constraint equations linear with respect to this highest coordinate derivative. Without loss of generality we assume that "*p*" stands for the highest order derivative of a coordinate which appears linearly in a constraint equation. It is possible that constraints are linear with respect to only one $p$-th order coordinate derivative. However, for simplicity of the development, we assume constraints linear in all $p$-th order derivatives of coordinates. Based on our assumption, instead of (*) we may write the constraints in the form:

$$B(t, q, \dot{q}, ..., q^{(p-1)})q^{(p)} + s(t, q, \dot{q}, ..., q^{(p-1)}) = 0, \qquad (**)$$

where $B$ is a $(k \times n)$-dimensional matrix with $n > k$, and it is assumed to have full rank, and $s$ is a $(k \times 1)$-vector.

**(**) is the unified constraint formulation.**

Integrability conditions for (**) can be found in

Tarn T-J., Zhang M., Serrani A., New Integrability Conditions for Differential Constraints, Systems & Control Letters, 49, pp. 335-345, 2003.

**According to definitions 1, 2 and (**), driving and task constraints, performance goals or other requirements on a system motion to obtain its specified performance may be included into the "programmed constraints" class. They can get the unified name as they play the same role – they program the motion.**
**They may be treated in the same way in dynamics and control.**

**There are many possible programmed constraints. It has to be verified whether a programmed constraint formulated for a system is reachable for it. It can be done by inspection of solutions of equations of a programmed motion.**

# TRANSITION FROM THE CONSTRAINED SYSTEM TO THE CONTROL SYSTEM

**"Control theory would be quite sterile without concrete connections to the natural world. The process of modeling is just as central to control engineering as is control theory itself. A control system design project does not begin when a control engineer is handed a model; it begins at the onset of model formulation"**

## KINEMATIC CONTROL MODELS

Constraint equations can be presented in the control form by viewing the independent generalized velocities as inputs. Kinematic control models take the form

$$\dot{q}_j = u_j,$$

$$\dot{q}_i = \sum_{j=1}^{m} \varphi_{ij}(q_1,\ldots,q_n)u_j. \qquad i=m+1,\ldots,n$$

Often, the above nonlinear control form is presented as

$$\dot{y}_i = u_i,$$

$$\dot{z} = \sum_{i=1}^{m} \tilde{g}_i(z,y)u_i, \qquad m\geq 2$$

where $q$ is partitioned as $q=(z,y)$, and $z=(q_{m+1},\ldots,q_n)$ is a *(n-m)*-fiber vector, and $y=(q_1,\ldots,q_m)$ is a *m*- base vector.

If the Chaplygin assumption holds, kinematic control model of the Chaplygin system can be presented in the nonlinear control form as

$$\dot{q}_j = u_j,$$

$$\dot{q}_i = \sum_{j=1}^{m} \varphi_{ij}(q_1,\dots,q_m)u_j. \qquad i=m+1,\dots,n$$

The control system is Chaplygin if $\varphi_{ij}$ depend on the base vector but not on the fiber vector. Most systems with material nonholonomic constraints are Chaplygin and for this reason the latter kinematic control model is a focus of many theoretic control studies. The Chaplygin kinematic control systems can be presented in the so-called chained or power forms. Since chained and power forms are used to model nonholonomic systems of practical importance such as front-wheel drive vehicles, multibody spacecraft, or tractors with trailers, it is no surprise that many studies are focused on these classes of systems.

The high order constraint equations (**)

$$B(t,q,\dot{q},\dots,q^{(p-1)})q^{(p)} + s(t,q,\dot{q},\dots,q^{(p-1)}) = 0,$$

can be transformed into the state space representation. To this end, we introduce a new $p$-vector $x = (x_1,\dots,x_p)$ such that $x_1 = q, \dot{x}_1 = x_2,\dots,\dot{x}_{p-1} = x_p$. We assume that $t$ is not present explicitly in (*). If it is, we reorder coordinates, assigning $x_0 = t$. With the new vector $x$ (**) can be written as $(p-1+k)$ first order equations

$$
\begin{aligned}
\dot{x}_1 &= x_2 , \\
\dot{x}_2 &= x_3 , \\
\vdots \quad & \quad \vdots \\
\vdots \quad & \quad \vdots \\
\dot{x}_{p-1} &= x_p , \\
B( x_1 ,..., x_p )\dot{x}_p &= -s( x_1 ,..., x_p )
\end{aligned}
$$

or

$$C(x)\dot{x} = b(x),$$

where $C$ is a $(p\text{-}1+k)\times p$ matrix and $b$ is a $(p\text{-}1+k)$- dimensional vector. Let $f(x)$ be a particular solution so $C(x)f(x) = b(x)$. Let $g(x)$ be a $p\times(n\text{-}k)$ full rank matrix whose column space is in the null space of $C(x)$, i.e. $C(x)g(x) = 0$. Then, the solution is given by

$$\dot{x} = f(x) + g(x)u(t)$$

for any smooth vector $u(t)$. The problem with the constraints (**) has been converted into a control problem for a system with high order constraints. In general, a drift term is present and the kinematic control model may be non-Chaplygin. It is the state space representation of the unified constraint formulation and we refer to it as a unified state space control formulation.

This kinematic control model is only formally equivalent to the standard model

$$\dot{x} = f( x ) + \sum_{i=1}^{m} g_i ( x )u_i ,$$

since $u_i , i = 1,...,m$, may not have a physical interpretation of velocities. They may be accelerations or their time derivatives.

# DYNAMIC CONTROL MODELS

Dynamic control models actually used, which we refer to as classical dynamic control models, are based on Lagrange's equations with multipliers

$$M(q)\ddot{q} + C(q,\dot{q}) + D(q) = J^T(q)\lambda + E(q)\tau,$$
$$J(q)\dot{q} = 0$$

where $M(q)$ is a $(n \times n)$ positive definite symmetric inertia matrix, $J(q)$ is a full rank $(k \times n)$ matrix, $2 \le n - k < n$, $\lambda$ is a $k$-dimensional vector of Lagrange's multipliers, $E(q)\tau$ is a $n$-dimensional vector of generalized forces applied to a system, and $\tau$ is a $r$-dimensional vector of control inputs.

For **model-based control applications**, the dynamic control model is to be transformed to the reduced-state form. It can be accomplished in several ways, e.g. we may start from Lagrange's equations with multipliers which we write as

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} = J^T(q)\lambda + Q(q,\dot{q}),$$
$$J(q)\dot{q} = 0,$$

where we assume that $Q(q,\dot{q})$ stands for all external forces applied to a system.

To eliminate constraint forces we project the equations onto the linear subspace generated by the null space of $J(q)$. Since $(J^T(q)\lambda) \cdot \delta q = 0$ Lagrange's equations become

$$\left[\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}}\right) - \frac{\partial T}{\partial q} - Q\right] \cdot \delta q = 0$$

where $\delta q \in R^n$ and satisfies $J(q)\delta q = 0$.

Partition $q$ and $J(q)$ such that $q = (q_1, q_2) \in R^{n-k} \times R^k$, $J = [J_1(q)\ J_2(q)]$, $J_2(q) \in R^{k \times k}$ is invertible, and $\delta q_2 = -J_2^{-1}(q)J_1(q)\delta q_1$, yields

$$\left[\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_1}\right) - \frac{\partial T}{\partial q_1} - Q_1\right] - J_1^T J_2^{-T}\left[\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{q}_2}\right) - \frac{\partial T}{\partial q_2} - Q_2\right] = 0.$$

These are the second order differential equations in terms of $q$.
They can be simplified by reusing the constraint equation $\dot{q}_2 = -J_2^{-1}(q)J_1(q)\dot{q}_1$ to eliminate $\dot{q}_2$ and $\ddot{q}_2$. Adding inputs $\tau_i$ we obtain the dynamic control model.
The evolution of $q_2$ can be retrieved by reapplication of the constraint equations. The control part of the constrained dynamics is

$$M(q)^* \ddot{q}_1 + C^*(q, \dot{q}_1)\dot{q}_1 + D^*(q) = E^*(q)\tau,$$
$$\dot{q} = G(q)\dot{q}_1.$$

The dynamic control model based on Lagrange's equations can be transformed to the state space form obtained by an extension of the kinematic control model as

$$\dot{q} = g(q)v = g_1(q)v_1 + ... + g_{n-k}(q)v_{n-k}, \qquad i=1,...n-k,\ 2 \le n-k < n$$

$$v_i^{r_i} = u_i,$$

where $r_i,...,r_m$ denote an order of time differentiation and $v$ is the output of a linear system consisting of chains of integrators.
This model is referred to as a **dynamic control model** since in applications from mechanics $r_i = 1, i = 1,...,n-k$, controls are typically generalized forces.
The model consists of the constraint equation and the dynamic equations of motion, which reduce to $\dot{v} = u$.
Indeed, equation $J(q)\dot{q} = 0$ constrains the velocity $\dot{q}$ at each $q$ to the null space of $J(q)$. Let the vector fields $g_1,...,g_m, m = n-k,$ form the basis for the null space of $J(q)$ at each $q$, and let $g(q) = (g_1(q),...,g_m(q))$.
Then $J(q)g(q) = 0$ for each $q$ and the constraint equations can be presented as

$$\dot{q} = g(q)v = g_1(q)v_1 + ... + g_{n-k}(q)v_{n-k},$$

for some appropriately defined $m$-dimensional vector $v = (v_1,...,v_m)$. Components of $v$ may or may not have physical interpretations as velocities.
By differentiating $\dot{q} = g(q)v$ we obtain

$$\ddot{q} = g(q)\dot{v} + \dot{g}(q)v.$$

Substituting the above into the motion equations and premultiplying by $g^T(q)$ we obtain

$$g^T(q)M(q)g(q)\dot{v} + F(q,\dot{q}) = g^T(q)E(q)\tau,$$

in which $F(q,\dot{q}) = g^T(q)[M(q)\dot{g}(q)v + C(q,\dot{q}) + D(q)]$.

We assume that the map $g^T(q)E(q)$ is onto what means that we require that independent degrees of freedom of the system are actuated.

Then, applying feedback linearization $U(\dot{q},q,u): R^n \times R^n \times R^m$ such that $\dot{v} = u$, $u = (u_1,...,u_m)$ is a $m$-vector control, we obtain

$$\dot{v}_i = u_i.$$

A class of dynamic nonholonomic systems for which the vector fields have a special form has been studied. The unicycle belongs to this class, for example. For this class of systems, it is possible to select a basis vector field for the null space of $J(q)$ such that $g(q)$ can be

$$g(q) = \begin{bmatrix} \tilde{g}(q) \\ I_k \end{bmatrix},$$

where $\tilde{g}(q)$ is a $k \times (n\text{-}k)$ matrix and $I_k$ is a $(n\text{-}k) \times (n\text{-}k)$ identity matrix. Partition of $q$ as $q = (z, y)$, where $z = (z_1, \ldots, z_k)$, $y = (y_1, \ldots, y_{n-k})$ results in the dynamic extension of the kinematic control model, which is

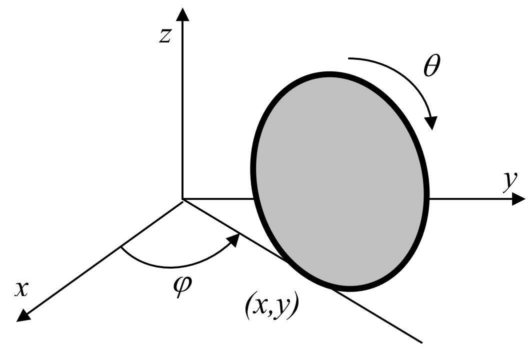
$$\ddot{y}_i = u_i, \qquad i=1,\ldots,n\text{-}k$$

$$\dot{z} = \sum_{i=1}^{n-k} \tilde{g}_i(z, y)\dot{y}_i.$$

These equations are said to be in the dynamic Chaplygin form if $\tilde{g}_i(q)$, $i=1,\ldots,n\text{-}k$, depend only on the base vector $y$ but not on the fiber vector $z$.

# EXAMPLES OF CONSTRAINED SYSTEMS

**Example 1**
**A unicycle model**




A coordinate vector $q = (x, y, \varphi, \theta) \in \Omega, \quad \Omega = R^2 \times SO^1 \times SO^1$
The wheel rolls without slipping

$$\dot{x} = \dot{\theta} r \cos\varphi, \quad \dot{y} = \dot{\theta} r \sin\varphi.$$

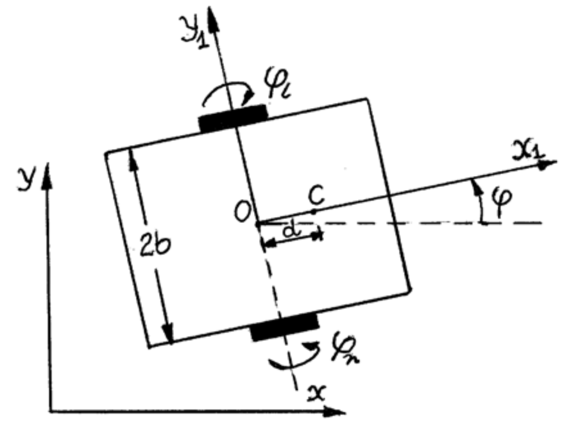
The kinematic control model:
Select 3-1=2 control inputs $\dot{\theta} r = v = u_1, \quad \dot{\varphi} = \omega = u_2,$
Then the constraint equations can be transformed to

$$\dot{x} = v\cos\varphi,$$
$$\dot{y} = v\sin\varphi,$$
$$\dot{\varphi} = \omega.$$

and $\dot{q} = \sum_{i=1}^{m} g_i(q) u_i$, has the form

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos\varphi \\ \sin\varphi \\ 0 \end{bmatrix} u_1 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u_2.$$

**Example 2**
**Two-wheeled mobile platform**




a coordinate vector $q = (x, y, \varphi, \varphi_R, \varphi_L) \in \Omega, \quad \Omega = R^2 \times SO^3$

The material constraint equations:

$$\dot{y}_c \cos\varphi - \dot{x}_c \sin\varphi - \dot{\varphi} d = 0,$$
$$\dot{x}_c \cos\varphi + \dot{y}_c \sin\varphi + \dot{\varphi} b = r\dot{\varphi}_r,$$
$$\dot{x}_c \cos\varphi + \dot{y}_c \sin\varphi - \dot{\varphi} b = r\dot{\varphi}_l.$$

Not all constraints are nonholonomic; one is holonomic, i.e.

$$\varphi = \frac{r}{2b}(\varphi_r - \varphi_l) + c_1.$$

For a real vehicle we can control $\omega_r(\dot{\varphi}_r),\ \omega_l(\dot{\varphi}_l)$ (or moments)

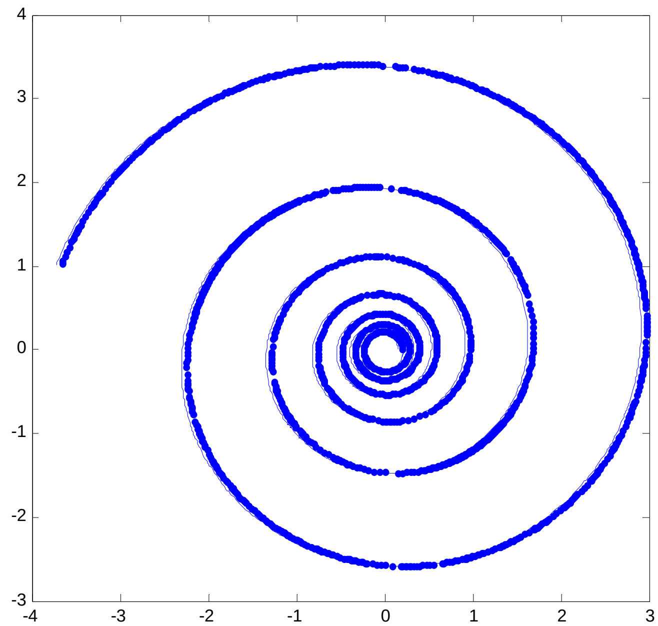
$$u_1 = \frac{r(\dot{\varphi}_r + \dot{\varphi}_l)}{2}, \quad u_2 = \frac{r(\dot{\varphi}_r - \dot{\varphi}_l)}{2d}.$$

Additionally, formulate the programmed constraint equation for the robot motion i.e. we want it to move along a trajectory described by the equation

$$x_c^2 + y_c^2 = R_p^2 \quad \text{and} \quad R_p = 0.01t + 0.2.$$

The constraints put on the robot motion are both material and non-material. Both constraints can be presented in the unified constraint formulation (**).

Model-based tracking (model-based tracking control strategy for programmed motion) using the Wen-Bayard control law



Take the programmed constraint on the rate of change of the trajectory curvature:

$$\Phi_1 = F_0 + \frac{\dot{x}\dddot{y} - \dddot{x}\dot{y}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}$$

and $\Phi_1 = 2\sin t + 1$,

$F_0$ does not contain third order coordinate derivatives.

[Derivation:

$$\Phi(t) = \frac{\begin{vmatrix} \dot{x} & \dot{y} \\ \ddot{x} & \ddot{y} \end{vmatrix}}{(\dot{x}^2 + \dot{y}^2)^{3/2}}$$

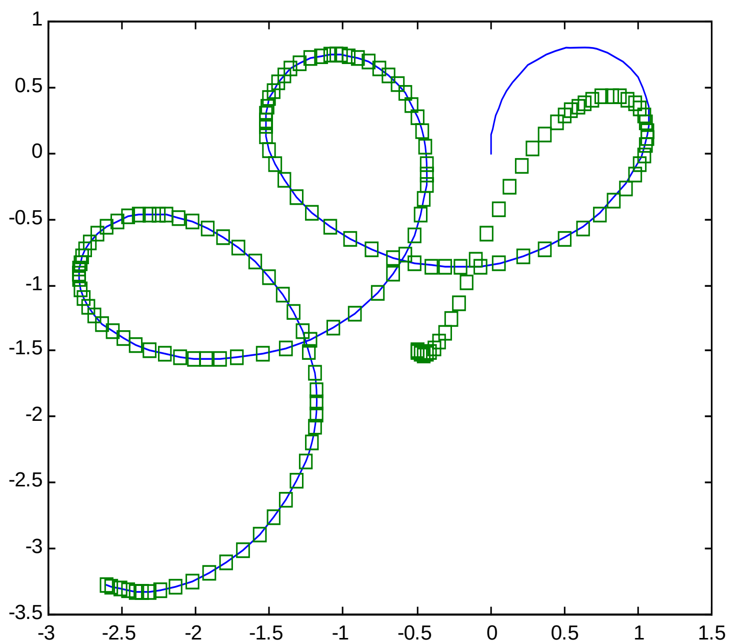
To specify a constraint on the rate of change of the curvature profile $\dot{\Phi}(t)$ this equation has to be differentiated

$$\dddot{x} = \frac{-\Phi(\dot{x}^2 + \dot{y}^2)^2\left[\dot{\Phi}(\dot{x}^2 + \dot{y}^2) + 3\Phi(\dot{x}\ddot{x} + \dot{y}\ddot{y})\right]}{\dot{y}(\dot{x}\ddot{y} - \ddot{x}\dot{y})} + \dddot{y}\frac{\dot{x}}{\dot{y}}$$

or

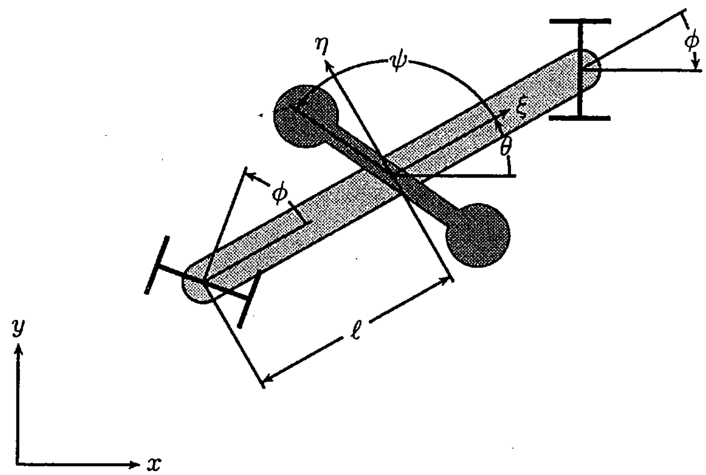
$$\dddot{x} = \frac{-\dot{\Phi}(\dot{x}^2 + \dot{y}^2)^{3/2}}{\dot{y}} - \frac{3(\dot{x}\ddot{x} + \dot{y}\ddot{y})(\dot{x}\ddot{y} - \dot{y}\ddot{x})}{\dot{y}(\dot{x}^2 + \dot{y}^2)} + \dddot{y}\frac{\dot{x}}{\dot{y}}.]$$

We assumed there were no external forces acted on the robot.
Computed torque control algorithm



**Example 3**

# Snake board – a system with idle wheels




**The snake board** performs undulatory motion.

$q = (x, y, \theta, \psi, \varphi) \in \Omega,\ \Omega = SE(2) \times S^1 \times S^1,$
$(x, y, \theta) -$ position of the snake board,
$(\psi, \varphi) -$ shape variables.
The constraint equations

$$-\dot{x}\sin(\theta+\varphi) + \dot{y}\cos(\theta+\varphi) - \dot{\theta}l\cos\varphi = 0,$$
$$-\dot{x}\sin(\theta-\varphi) + \dot{y}\cos(\theta-\varphi) + \dot{\theta}l\cos\varphi = 0.$$

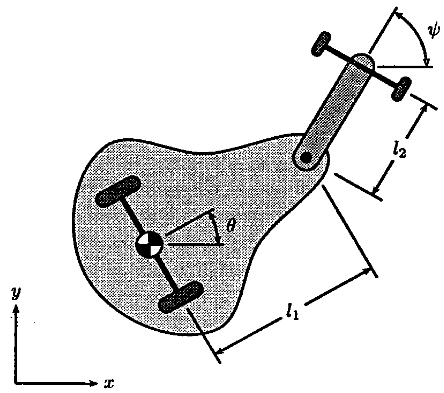
Control torques $\tau_\psi$, $\tau_\varphi$:

- Controlling body motion - $\tau_\psi$,
- Controlling wheels (by appropriate motion of feet) - $\tau_\varphi$.

There are 5 coordinates, 2 constraint equations, 2 control inputs

**Example 4**

# Roller - racer – a system with idle wheels




Motion of the roller-racer is described by $q=(x,y,\theta,\psi)\in\Omega$, $\Omega = SE(2)\times S^1$, where $(x,y,\theta)$ describe a position, and $\psi$ is a shape variable.

Platforms are connected with a rotary joint. The propulsion and steering come from a rotary motion at this joint.

The roller-racer performs undulatory motion.

A torque $\tau$ applied as this joint is the only control input.

Equations of nonholonomic constraints come form the assumption that the wheels do not slip, i.e.

$$\dot{x}\sin\theta - \dot{y}\cos\theta = 0,$$

$$-\dot{x}\sin\psi + \dot{y}\cos\psi + \dot{\theta}l_1\cos(\theta-\psi) + l_2\dot{\psi} = 0.$$

One control torque $\tau_\psi$ (4 coordinates, 2 equations of constraints, 1 control input)

The fundamental means of its propulsion is the pivoting of the steering handlebar around the joint axis and the nonholonomic constraints. The purely kinematic analysis of the roller-racer is not allowed. We cannot determine its global motion by just the shape variation, since it does not posses a sufficient number of nonholonomic constraint equations for this. Kinematics must be complemented with the system dynamics.

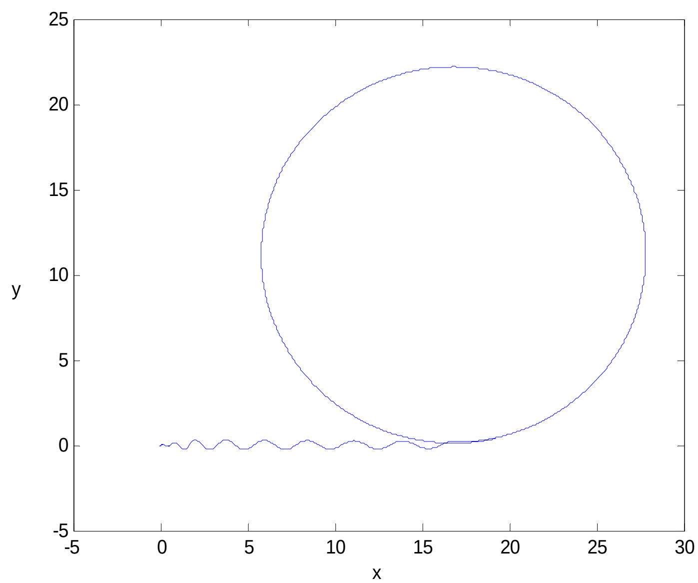
The system is underactuated.

# An example of programmed motion tracking:

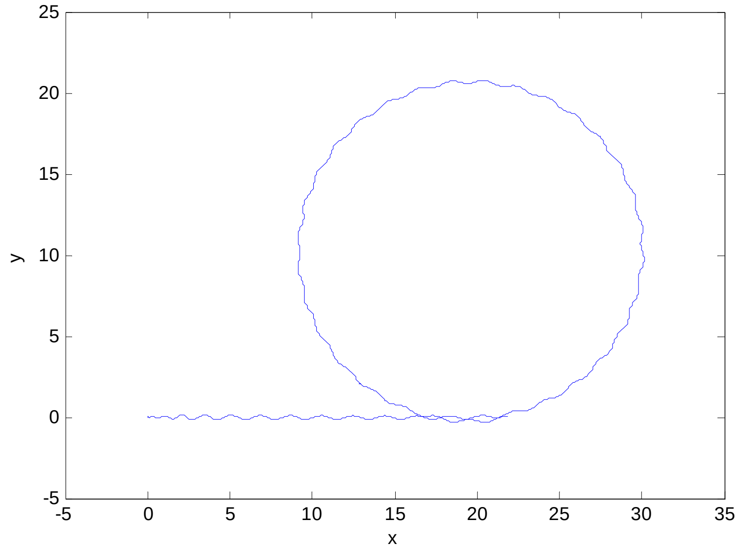
Jarzębowska E.: Lewandowski R.: Modeling and Control Design using the Boltzmann-Hamel Equations: A Roller-Racer Example. *Proc. 8th Intern. IFAC Symposium on Robot Control*, SYROCO 2006, Bologna, Italy, 2006.

We developed the control dynamics for the roller-racer using the Boltzmann-Hamel equations. Then, we applied a static state feedback linearization and designed a computed torque controller to track a desired maneuver for a roller-racer. This maneuver consists of driving a circular trajectory with some desired velocity. A rider learns what value of this velocity is enough to start a "smooth" maneuver, and changes his orientation with respect to the world coordinates so we parameterize the desired trajectory in terms of $\theta$ .

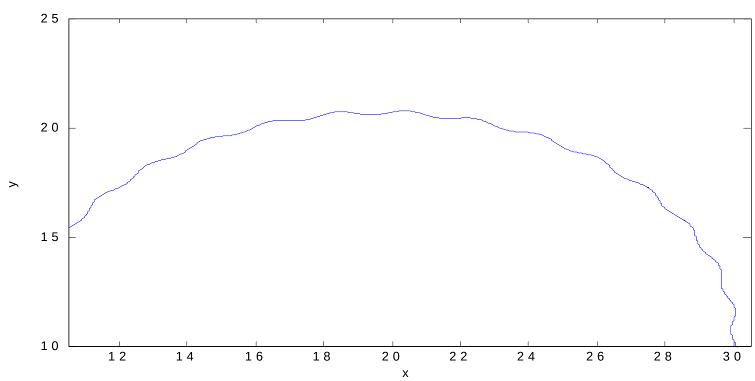
For illustrative purposes, the desired $\theta$ is given as $\theta_d = 0.2(t - t_1)$ where $t_1$ is the time at which a maneuver can start. We design a controller with respect to the quasi-velocity $\omega_2 = \dot{\theta}$.



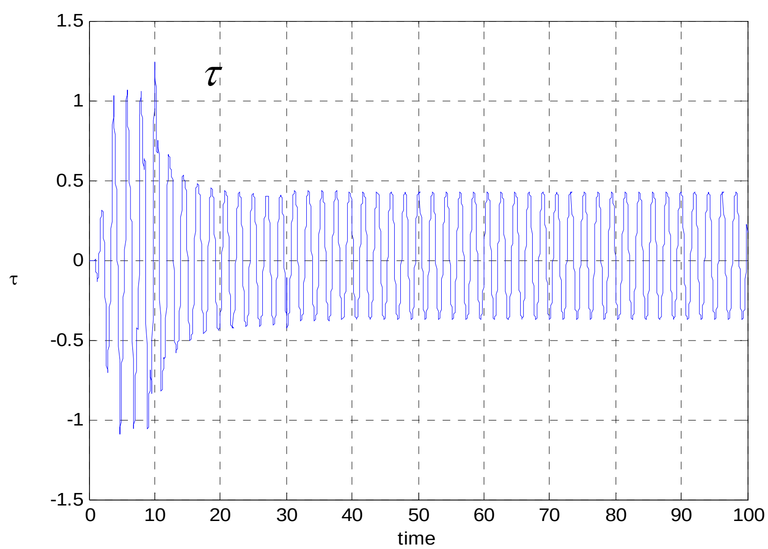

Desired trajectory tracking (no friction).




Desired trajectory tracking (with friction).




Magnification of the desired trajectory tracking

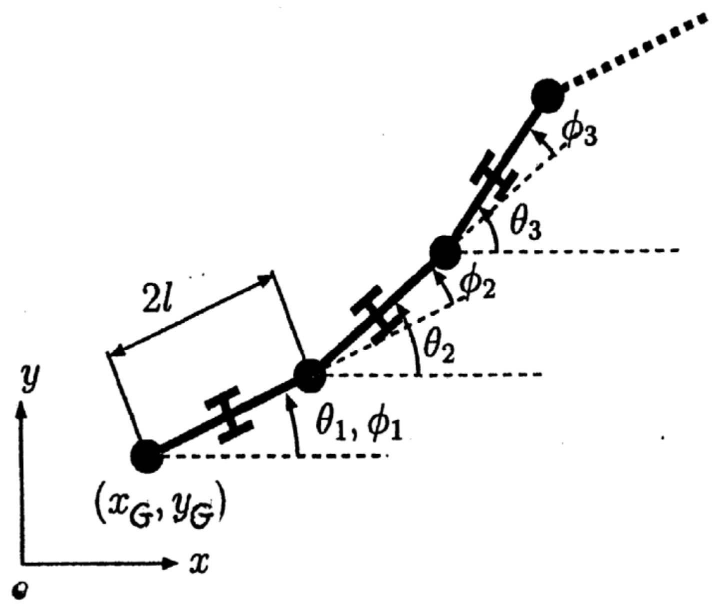



Control torque vs. time (with friction)

**Example 5**

## Snake – like robot

Roller - racer was the shortest snake – take a longer one

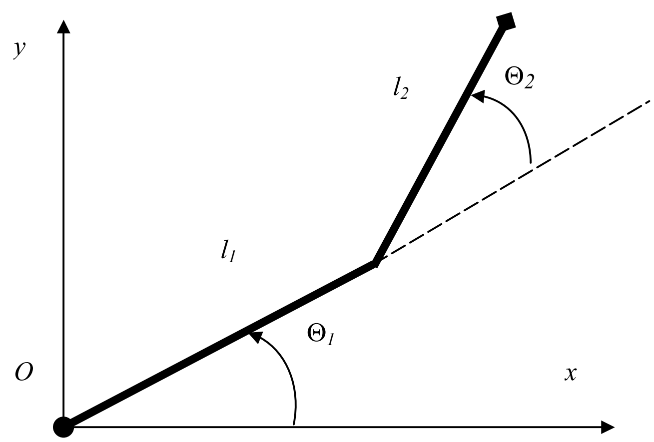



The constraint equations are the same as for other wheeled vehicles – one constraint equation per segment.
Kinematics must be complemented by the system dynamics.

**Example 6**

# A two-link planar manipulator model


Jarzębowska E.: Control Oriented Dynamic Formulation of Robotic Systems with Program Constraints. *Robotica*, **24**,1:61-73, 2006.
Jarzębowska E.: Tracking Control Design for Underactuated Constrained Systems. *Robotica*, **24**,1: 591-593, 2006.


The planar two-link manipulator may move in the horizontal plane *(x,y)*. Two degrees of freedom are described by $\Theta_1, \Theta_2$.

The system is holonomic itself.
The kinematic control model is trivial; the dynamic control model is not! This is a nonlinear control model for holonomic systems.

The programmed constraint may b formulated for the manipulator end-effector. It may specify writing, painting, scribing, e.t.c.
Formulate a requirement that the manipulator end-effector is to move along a trajectory for which its curvature changes according to a specified function $\Phi^* = \dfrac{d\Phi(t)}{dt}$.
For tracking control purposes the constraint equation is transformed from the task space coordinates *(x,y)* into the joint coordinates $(\Theta_1, \Theta_2)$ by inserting $x = l_1 \cos\Theta_1 + l_2 \cos(\Theta_1 + \Theta_2)$ and $y = l_1 \sin\Theta_1 + l_2 \sin(\Theta_1 + \Theta_2)$, and their time derivatives into the constraint equation. We obtain

$$F_2\ddot{\Theta}_1 + \ddot{\Theta}_2 - F_1 = 0,$$

where $F_1 = \dfrac{A_\phi - A_1 - A_2 a_o}{a_2 + a_4 a_o}$, $F_2 = \dfrac{a_1 + a_2 + a_o(a_3 + a_4)}{a_2 + a_4 a_o}$, and

$$A_\phi = \frac{-\Phi(a_5^2 + a_6^2)^2\left[\dot{\Phi}(a_5^2 + a_6^2) + 3\Phi(a_5 a_7 + a_6 a_8)\right]}{a_6(a_5 a_8 - a_7 a_6)},$$

$$A_1 = 3a_3\dot{\Theta}_1\ddot{\Theta}_1 + 3a_4(\ddot{\Theta}_1 + \ddot{\Theta}_2)(\dot{\Theta}_1 + \dot{\Theta}_2) - a_1\dot{\Theta}_1^{\;3} - a_2(\dot{\Theta}_1 + \dot{\Theta}_2)^3,$$

$$A_2 = 3a_3\dot{\Theta}_1\ddot{\Theta}_1 + 3a_2(\ddot{\Theta}_1 + \ddot{\Theta}_2)(\dot{\Theta}_1 + \dot{\Theta}_2) + a_3\dot{\Theta}_1^{\;3} + a_4(\dot{\Theta}_1 + \dot{\Theta}_2)^3,$$

$$a_1 = -l_1 \sin\Theta_1,\; a_2 = -l_2 \sin(\Theta_1 + \Theta_2),$$

$$a_3 = -l_1 \cos\Theta_1,\; a_4 = -l_2 \cos(\Theta_1 + \Theta_2),$$

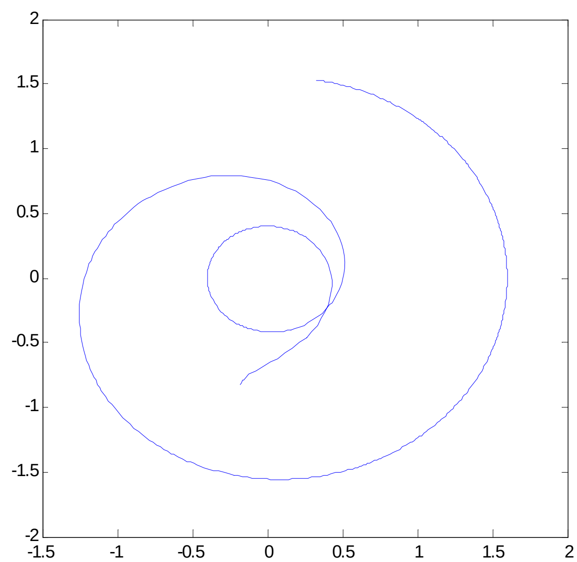
$$a_5 = a_1\dot{\Theta}_1 + a_2(\dot{\Theta}_1 + \dot{\Theta}_2),\; a_6 = -a_3\dot{\Theta}_1 - a_4(\dot{\Theta}_1 + \dot{\Theta}_2),$$

$$a_7 = a_1\ddot{\Theta}_1 + a_3\dot{\Theta}_1^2 + a_2(\ddot{\Theta}_1 + \ddot{\Theta}_2) + a_4(\dot{\Theta}_1 + \dot{\Theta}_2)^2,$$

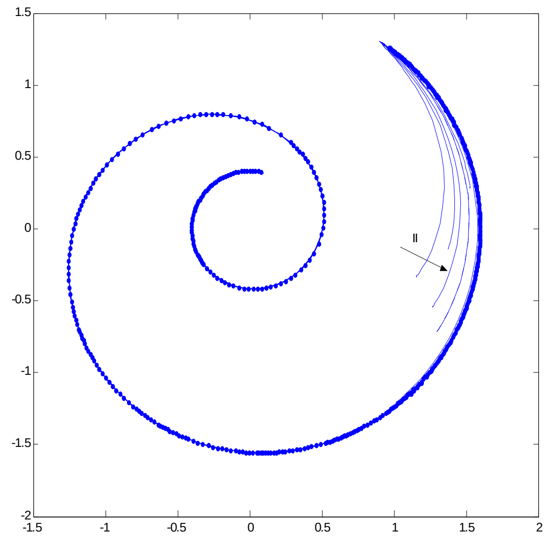
$$a_8 = -a_3\ddot{\Theta}_1 + a_1\dot{\Theta}_1^2 - a_4(\ddot{\Theta}_1 + \ddot{\Theta}_2) + a_2(\dot{\Theta}_1 + \dot{\Theta}_2)^2,$$

$$a_o = a_5 / a_6.$$

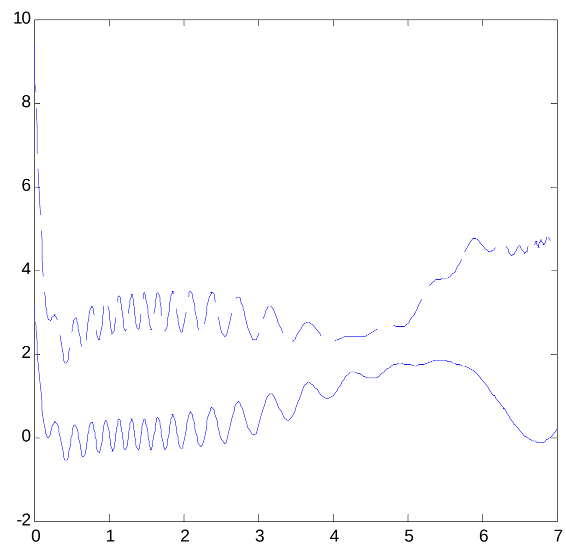
For $\Phi_1 = 0.6 + 0.02t$, applying the computed torque controller, the programmed motion looks like



# Tracking by the computed torque

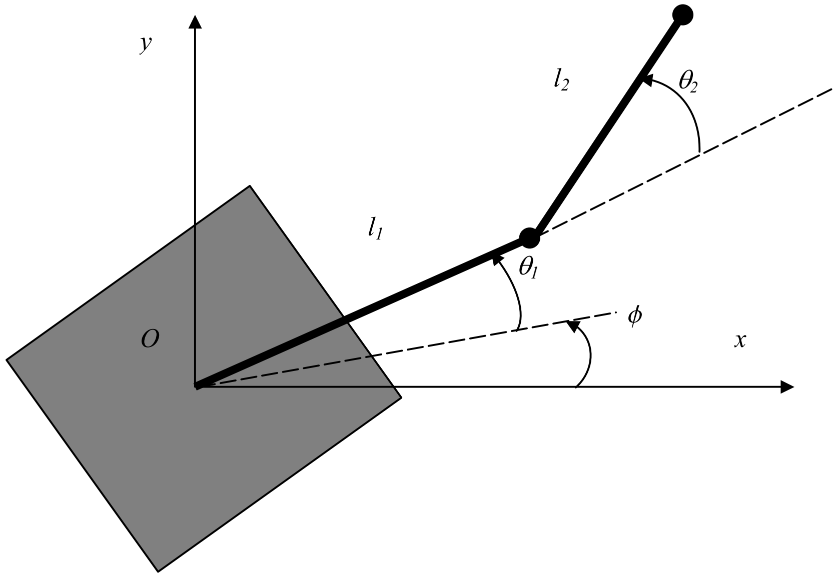



# Torques applied to the joints are



**EXAMPLE 7**

# AN EQUATION OF A PROGRAMMED CONSTRAINT FOR A SPACE TWO-LINK MANIPULATOR




Take a free-floating two-link manipulator that consists of a base described by its moment of inertia $J$ and $\phi$, which is the orientation of the base relative to a fixed axis in the plane. Let $\theta_1$ be the angle of the first link of mass $m_1$ and length $l_1$ relative to the base, and $\theta_2$ be the angle of the second link of mass $m_2$ and length $l_2$ relative to the first one. For simplicity we assume that link masses are concentrated at the ends of the links.

In our model the manipulator base is pinned to the ground at its center. Pinning the base permits the body to rotate freely but prevents translation.

For the manipulator **the law of conservation of the angular momentum implies that moving the links causes the base body to rotate.**

The conservation law is viewed as a nonholonomic constraint on the system and it has the form

$$\left[J+(m_1+m_2)l_1^2+m_2l_2^2\right]\dot{\phi}+\left[(m_1+m_2)l_1^2+m_2l_2^2\right]\dot{\theta}_1+$$
$$+m_2l_2^2\dot{\theta}_2+m_2l_1l_2\cos\theta_2(2\dot{\phi}+2\dot{\theta}_1+\dot{\theta}_2)=0.$$

The holonomic constraint arising from the linear momentum conservation in a real space manipulator is replaced with holonomic pinned constraints in our model. Pinning the base body simplifies the model but does not remove any of its essential structure.

Additionally, we wish to add a position programmed constraint, that specifies motion along a straight line for the end of the second link. This constraint has the form

$$l_1\left[\sin(\theta_1+\phi)-\alpha\cos(\theta_1+\phi)\right]+$$
$$+l_2\left[\sin(\theta_1+\theta_2+\phi)-\alpha\cos(\theta_1+\theta_2+\phi)\right]-\beta=0,$$

where $\alpha$ and $\beta$ specify the line position.

Altogether, we have the nonholonomic constraint linear in velocities and a holonomic constraint. Both constraint equations can be transformed to the unified constraint form as follows:

$$B_{1\varphi}\dot{\varphi}+B_{1\theta1}\dot{\theta}_1+B_{1\theta2}\dot{\theta}_2=0,$$

where

$$B_{1\varphi}=J+(m_1+m_2)l_1^2+m_2l_2^2+2m_2l_1l_2\cos\theta_2,$$
$$B_{1\theta1}=(m_1+m_2)l_1^2+m_2l_2^2+2m_2l_1l_2\cos\theta_2,$$
$$B_{1\theta2}=m_2l_2^2+m_2l_1l_2\cos\theta_2,$$

and differentiate the programmed constraint with respect to time

$$A_{1\varphi}\dot{\varphi} + A_{1\theta 1}\dot{\theta}_1 + A_{1\theta 2}\dot{\theta}_2 + \sigma A_1 = 0,$$

where

$$A_1 = l_1[sin(\theta_1+\varphi) - \alpha\, cos(\theta_1+\varphi)] + l_2[sin(\theta_1+\theta_2+\varphi) - \alpha\, cos(\theta_1+\theta_2+\varphi)] - \beta = 0,$$
$$A_{1\varphi} = l_1\, cos(\theta_1+\varphi) + l_1\alpha\, sin(\theta_1+\varphi) + l_2\, cos(\theta_1+\theta_2+\varphi) + l_2\alpha\, sin(\theta_1+\theta_2+\varphi),$$
$$A_{1\theta 1} = l_1\, cos(\theta_1+\varphi) + l_1\alpha\, sin(\theta_1+\varphi) + l_2\, cos(\theta_1+\theta_2+\varphi) + l_2\alpha\, sin(\theta_1+\theta_2+\varphi),$$
$$A_{1\theta 2} = l_2\, cos(\theta_1+\theta_2+\varphi) + l_2\alpha\, sin(\theta_1+\theta_2+\varphi),$$

$\sigma$ is a positive constant that has to be selected to stabilize the constraint for simulation.

All equations of constraints for the space manipulator are

$$\begin{bmatrix} A_{1\varphi} & A_{1\theta 1} & A_{1\theta 2} \\ B_{1\varphi} & B_{1\theta 1} & B_{1\theta 2} \end{bmatrix} \begin{bmatrix} \dot{\varphi} \\ \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} = \begin{bmatrix} -\sigma A_1 \\ 0 \end{bmatrix}.$$

**EXAMPLE 8**

# The two-link manipulator model that is underactuated.

Jarzębowska E.: Tracking Control Design for Underactuated Constrained Systems. *Robotica*, **24**,1: 591-593, 2006.

Remove one control input (its broken) in the two-link planar manipulator - it is equipped with one actuator.
Formulate the following tracking control objective: Its end-effector is to move according to a specified programmed motion and it is the only constraint put upon the manipulator. The first joint is actuated and the second is not.

The manipulator control model with the first joint actuated is

$$\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}\begin{bmatrix} \ddot{\Theta}_1 \\ \ddot{\Theta}_2 \end{bmatrix} + \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix}\begin{bmatrix} \dot{\Theta}_1 \\ \dot{\Theta}_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ 0 \end{bmatrix},$$

where the equation for the unactuated joint is a second order nonholonomic constraint equation (the variable $\Theta_2$ is present in the inertia matrix). The manipulator is a holonomic system for which the following transformations may be obtained

$$\ddot{\Theta}_2 = -\frac{1}{\delta}\left[(\delta + \beta \cos\Theta_2)\ddot{\Theta}_1 + \beta \sin\Theta_2 \dot{\Theta}_1^2\right]$$

and

$$\left[(\alpha + 2\beta\cos\Theta_2) - \frac{1}{\delta}(\delta + \beta\cos\Theta_2)^2\right]\ddot{\Theta}_1 -$$

$$+\dot{\Theta}_2 \beta \sin\Theta_2 (2\dot{\Theta}_1 + \dot{\Theta}_2) - \frac{\dot{\Theta}_1^2 \beta \sin\Theta_2}{\delta}(\delta + \beta\cos\Theta_2) = \tau_1.$$

Using the partial feedback linearizing controller

$$\tau_1 = \left[ (\alpha + 2\beta \cos\Theta_2) - \frac{1}{\delta}(\delta + \beta\cos\Theta_2)^2 \right] u -$$
$$+ \left[ \dot{\Theta}_2 \beta \sin\Theta_2 (2\dot{\Theta}_1 + \dot{\Theta}_2) + \frac{\dot{\Theta}_1^2 \beta \sin\Theta_2}{\delta}(\delta + \beta\cos\Theta_2) \right],$$

we obtain

$$\ddot{\Theta}_1 = u,$$
$$\ddot{\Theta}_2 = -\frac{1}{\delta}(\delta + \beta\cos\Theta_2)\ddot{\Theta}_1 - \frac{1}{\delta}\beta\sin\Theta_2\dot{\Theta}_1^2.$$

These equations can be expressed in the state space control formulation by defining the following state variables: $x_1 = \Theta_1, x_2 = \Theta_2, x_3 = \dot{\Theta}_1, x_4 = \dot{\Theta}_2$. Then, we obtain

$$\dot{x}_1 = x_3,$$
$$\dot{x}_2 = x_4,$$
$$\dot{x}_3 = u,$$
$$\dot{x}_4 = -\frac{x_3^2 \beta \sin x_2}{\delta} - \frac{1}{\delta}(\delta + \beta\cos x_2)u$$

which is

$$\dot{x} = f(x) + g(x)u,$$

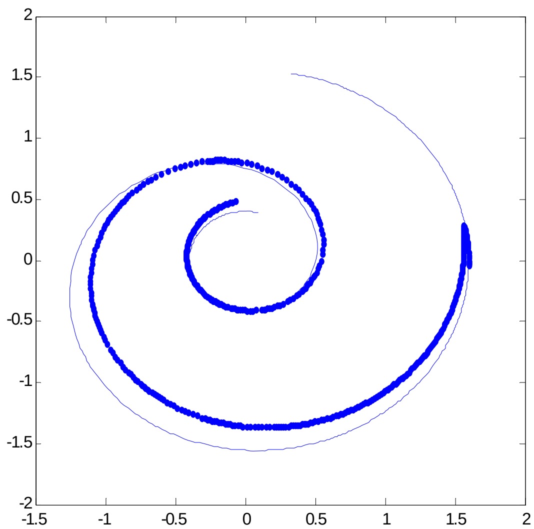
where

$f(x) = (x_3, x_4, 0, -x_3^2 \beta \sin x_2 / \delta),$

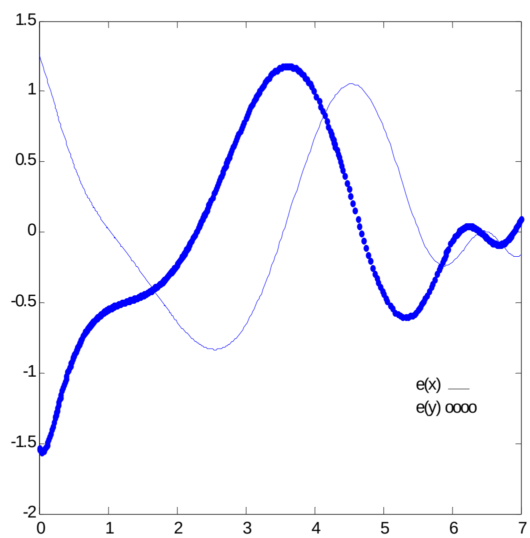
$g(x) = (0, 0, e_1, -(\delta + \beta\cos x_2)/\delta)$

with $e_1$ - the standard basis vector in $R^1$ are the drift and control vector fields on $\Omega = (-\pi/2, \pi/2) \times (-\pi/2, \pi/2) \times R^2$.

Select the programmed constraint for the rate of change of the trajectory curvature and apply the PD controller.
Tracking by the PD controller




Time histories of position tracking errors



# COMPUTATION PROBLEMS

**There are computation problems with nonholonomic systems.**

Holonomic constraints, either material or programmed, are often differentiated with respect to time for modeling and control purposes and they are written in the form

$$\frac{dA_1(t,q)}{dt} = A_{1t}(t,q) + A_{1q}(t,q)\dot{q} = 0,$$

These velocity-level constraints are equivalent to the position constraints provided that the initial condition for a system $q_0 = q(t_0)$ is a valid one, i.e. $A_1(t_0, q_0) = 0$. To use this form of position constraints in computer simulations one must secure that a system starts from a valid initial condition. It may mean that one must numerically solve a system of nonlinear equations $A_1(t,q) = 0$ for the valid initial $q_0$. If the initial condition is not valid, the holonomic constraints are violated in simulation and they cannot be enforced.
To remedy this problem, the differentiated constraints are replaced by

$$A_{1t}(t,q) + A_{1q}(t,q)\dot{q} + \sigma A_1(t,q) = 0.$$

The positive constant σ is a convergence rate of the modified constraints to the original ones, when initial conditions do not satisfy the original constraints. It can be appropriately chosen based on specific applications.

Often, position constraints have the form $A_1(q) = 0$ and kinematic constraints are linear in velocities, i.e. $B_1(t,q)\dot{q} + b_1(t,q) = 0$. Then, position and kinematic constraints written together, take the form

$$B_2^*(t,q)\dot{q} + b_2^*(t,q) = 0,$$

where

$$B_2^* = \begin{bmatrix} A_{1q}(q) \\ B_1(t,q) \end{bmatrix}, \qquad b_2^* = \begin{bmatrix} \sigma A_1(q) \\ b_1(t,q) \end{bmatrix},$$

and $B_2^*(t,q)$ is assumed to be a full rank $(a+k) \times n$ matrix and $b_2^*(t,q)$ is a $(a+k)$-dimensional vector.
This constraint form is equivalent to the unified constraint formulation for $p=1$.

The situation changes when both material and programmed constraints of different orders are merged into one dynamic model .........

# EXPERIMENTS IN NONHOLONOMICITY (BUILDING NONHOLONOMIC SYSTEMS)

There are more and more experiments with nonholonomic systems ……

- **ball on a rotating table**
- **nonholonomic manipulator** – to test control algorithms dedicated to nonholonomic systems, prototyping systems that consist of „nonholonomic modules",
- **air spindle test bed** – to test new control methods,
- **nonholonomic needle** – to test whether a nonholonomic model may describe other phenomena,
- desktop manipulator – **mobipulator – "a nonholonomic desk cleaner"**,

# CONCLUSIONS

Jarzębowska E.: Stabilizability and Motion Tracking Conditions for Mechanical Nonholonomic Control Systems, *Mathematical Problems in Engineering*, Hindawi Publ. Corp.. Vol. 2007

## Classification of nonholonomic constraints

| Kind of constraints | Systems / Constraint equations | Number of degrees of freedom ($m$), number of control inputs ($l$) | LAS | Tracking |
|---|---|---|---|---|
| 1. First order, material nonholonomic. | Car-like vehicles, mobile platforms with powered wheels, multi-fingered hands, nonholonomic manipulators, dexterous manipulation. $B_1(q,\dot{q})=0$ (1) $B_1$ is a $(k\times n)$ full rank matrix, $n>k$. | $m=n-k$ $m=l$ | - | + |
| | Wheeled vehicles with idle wheels, nonholonomic toys, snake-like robots and manipulators. Constraints have the form (1), $n>k$. | $m=n-k$ $m\geq l$ | - | + |
| 2. First order, non-material nonholonomic (conservation law). | Space vehicles and robots, sportsman, falling cat. $B_2(q)\dot{q}+b_2(q)=0$ (2) $B_2$ is a $(k\times n)$ full rank matrix, $n>k$ | $m=n-k$ $m\geq l$ | May be | + |

## Classification of nonholonomic constraints

| | | | | |
|---|---|---|---|---|
| 3. Second order, non-material nonholonomic, (underactuated). | Manipulators, space systems, underwater vehicles.<br>$M_{11}(q)\ddot{q}_1 + M_{12}(q)\ddot{q}_2 + C_1(q,\dot{q}) = T_1(q)\tau,$<br>$M_{21}(q)\ddot{q}_1 + M_{22}(q)\ddot{q}_2 + C_2(q,\dot{q}) = 0,$ (3a) | no gravity is present:<br>$m=n,$<br>$m>l$ | - | + |
| | $M_{11}(q)\ddot{q}_1 + M_{12}(q)\ddot{q}_2 + C_1(q,\dot{q}) + D_1(q) = T_1(q)\tau,$<br>$M_{21}(q)\ddot{q}_1 + M_{22}(q)\ddot{q}_2 + C_2(q,\dot{q}) + D_2(q) = 0,$ (3b) | gravity is present:<br>$m=n,$<br>$m>l$ | + | + |
| 4. High order, non-material nonholonomic (programmed). | Task specifications for any system:<br>$B(t,q,\dot{q},...,q^{(p-1)})q^{(p)} + s(t,q,\dot{q},...,q^{(p-1)}) = 0,$ (4)<br>$B$ is a $(k \times n)$ full rank matrix, $n \geq k$, $s$ is a $(k \times 1)$ vector. | $m=n-k,$<br>$m \geq l$ | May be | + |
| 5. Different types of constraints put on a system. | Underactuated vehicles with idle wheels, manipulators and other systems with material and programmed constraints.<br>The unified constraint (4), $n \geq k$. | $m=n-k,$<br>$m \geq l$ | May be | + |