Joint-Space Control

Optimization-based Robot Control

Andrea Del Prete

University of Trento
This document explains the control framework known as Task-Space Inverse Dynamics (TSID).

TSID is a popular control framework for legged robots.
This document explains the control framework known as **Task-Space Inverse Dynamics (TSID)**.

TSID is a popular control framework for legged robots.

It all started in 1987 with this paper by *Oussama Khatib*: “A unified approach for motion and force control of robot manipulators: The operational space formulation” [5]
This document explains the control framework known as **Task-Space Inverse Dynamics (TSID)**.

TSID is a popular control framework for legged robots.

It all started in 1987 with this paper by Oussama Khatib: “A unified approach for motion and force control of robot manipulators: The operational space formulation” [5]

Very active research topic between 2004 and 2015 [10, 6, 7, 9, 4, 3].
This document explains the control framework known as Task-Space Inverse Dynamics (TSID).

TSID is a popular control framework for legged robots.

It all started in 1987 with this paper by Oussama Khatib: “A unified approach for motion and force control of robot manipulators: The operational space formulation” [5]

Very active research topic between 2004 and 2015 [10, 6, 7, 9, 4, 3].

Now not so active anymore (i.e. problem solved), but widely used.
1. Theory of Joint Space Control (≈ 1:15 hour)
2. Implementation (≈ 1 hour)
3. Theory of Task Space Control (≈ 1:15 hour)
4. Implementation (≈ 1 hour)
The state of the system is denoted $x \triangleq (q, v)$.

**Configuration** vector $q \in \mathbb{R}^{n_q}$ of (relative) joint angles.

**Velocity** vector $v = \dot{q} \in \mathbb{R}^{n_v}$ of (relative) joint velocities.

The control inputs are denoted $u \triangleq \tau$ (joint torques).
The state of the system is denoted $\mathbf{x} \triangleq (q, v)$.

**Configuration vector** $q \in \mathbb{R}^{n_q}$ of (relative) joint angles.

**Velocity vector** $\mathbf{v} = \dot{q} \in \mathbb{R}^{n_v}$ of (relative) joint velocities.

The **control inputs** are denoted $u \triangleq \tau$ (joint torques).

The identity matrix is denoted $I$.

The zero matrix is denoted $0$.

When needed, the size of the matrix is written as index, e.g., $I_3$. 
| 1. Joint-Space Inverse Dynamics Control |
| 2. Inverse Dynamics Control as Optimization Problem |
Joint-Space Inverse Dynamics Control
Given (nonlinear) manipulator dynamics:

\[ M(q) \dot{v} + h(q, v) = \tau \]  \hspace{1cm} (1)

**Problem**
Find \( \tau(t) \) so that \( q(t) \) follows reference \( q^r(t) \).

Assumption
We know dynamics and can measure \( q \) and \( v \).

Solution
Set \( \tau = M(q) \dot{v} + h(q, v) \rightarrow \) closed-loop dynamics is \( \dot{v} = \dot{v}_d \).

Select \( \dot{v}_d \) so that \( q(t) \) follows \( q^r(t) \):

\[ \dot{v}_d = \dot{v}_r - K_d (v - v_r) - K_p (q - q_r) \]  \hspace{1cm} (2)

where \( K_p, K_d \) are diagonal positive-definite gain matrices.
Given (nonlinear) manipulator dynamics:

\[ M(q)\dot{v} + h(q, v) = \tau \]  \hspace{1cm} (1)

**Problem**

Find \( \tau(t) \) so that \( q(t) \) follows reference \( q^r(t) \).

**Assumption**

We know dynamics and can measure \( q \) and \( v \).
Given (nonlinear) manipulator dynamics:

\[ M(q) \ddot{v} + h(q, v) = \tau \]  

(1)

**Problem**

Find \( \tau(t) \) so that \( q(t) \) follows reference \( q^r(t) \).

**Assumption**

We know dynamics and can measure \( q \) and \( v \).

**Solution**

Set \( \tau = M(q) \dot{v}^d + h(q, v) \rightarrow \) closed-loop dynamics is \( \dot{v} = \dot{v}^d \).
Given (nonlinear) manipulator dynamics:

\[ M(q)\dot{v} + h(q, v) = \tau \]  

(1)

**Problem**

Find \( \tau(t) \) so that \( q(t) \) follows reference \( q^r(t) \).

**Assumption**

We know dynamics and can measure \( q \) and \( v \).

**Solution**

Set \( \tau = M(q)\dot{v}^d + h(q, v) \rightarrow \) closed-loop dynamics is \( \dot{v} = \dot{v}^d \).

Select \( \dot{v}^d \) so that \( q(t) \) follows \( q^r(t) \):
Given (nonlinear) manipulator dynamics:

\[
M(q)\dot{v} + h(q, v) = \tau
\]  

(1)

**Problem**

Find \( \tau(t) \) so that \( q(t) \) follows reference \( q^r(t) \).

**Assumption**

We know dynamics and can measure \( q \) and \( v \).

**Solution**

Set \( \tau = M(q)\dot{v}^d + h(q, v) \) \( \rightarrow \) closed-loop dynamics is \( \dot{v} = \dot{v}^d \).

Select \( \dot{v}^d \) so that \( q(t) \) follows \( q^r(t) \):

\[
\dot{v}^d = \dot{v}^r
\]
Robot Manipulator

Given (nonlinear) manipulator dynamics:

\[ M(q)\dot{v} + h(q, v) = \tau \]  \hspace{1cm} (1)

**Problem**

Find \( \tau(t) \) so that \( q(t) \) follows reference \( q^r(t) \).

**Assumption**

We know dynamics and can measure \( q \) and \( v \).

**Solution**

Set \( \tau = M(q)\dot{v}^d + h(q, v) \) \( \rightarrow \) closed-loop dynamics is \( \dot{v} = \dot{v}^d \).

Select \( \dot{v}^d \) so that \( q(t) \) follows \( q^r(t) \):

\[ \dot{v}^d = \dot{v}^r - K_d(v - v^r) - K_p(q - q^r) \]  \hspace{1cm} (2)

where \( K_p, K_d \) are diagonal positive-definite gain matrices.
Show that $q(t)$ converges to $q'(t)$. 
Convergence

Show that $q(t)$ converges to $q^r(t)$.

Closed-loop dynamics is

\[
\dot{v} = \dot{v}^r - K_d (v - v^r) - K_p (q - q^r)
\]

$A$ is Hurwitz if $K_p$ and $K_d$ are diagonal and positive-definite
Convergence

Show that $q(t)$ converges to $q^r(t)$.

Closed-loop dynamics is

$$\dot{v} = \dot{v}^r - K_d (v - v^r) - K_p (q - q^r)$$

$$\ddot{e} = -K_d \dot{e} - K_p e$$
Show that $q(t)$ converges to $q^r(t)$.

Closed-loop dynamics is

$$
\dot{v} = \dot{v}^r - K_d (v - v^r) - K_p (q - q^r)
$$

$$
\ddot{e} = -K_d \dot{e} - K_p e
$$

$$
\begin{bmatrix}
\dot{e} \\
\ddot{e}
\end{bmatrix} =
\begin{bmatrix}
0 & 1 \\
-K_p & -K_d
\end{bmatrix}
\begin{bmatrix}
e \\
\dot{e}
\end{bmatrix}
$$

$A$ is Hurwitz if $K_p$ and $K_d$ are diagonal and positive-definite.

$$
\lim_{t \to \infty} x(t) = 0 \\
\lim_{t \to \infty} q(t) = q^r(t)
$$
Convergence

Show that \( q(t) \) converges to \( q^r(t) \).

Closed-loop dynamics is

\[
\begin{align*}
\dot{v} &= \dot{v}^r - K_d (v - v^r) - K_p (q - q^r) \\
\ddot{e} &= -K_d \dot{e} - K_p e \\
\begin{bmatrix}
\dot{e} \\
\ddot{e}
\end{bmatrix}
&= 
\begin{bmatrix}
0 & I \\
-K_p & -K_d
\end{bmatrix}
\begin{bmatrix}
e \\
\dot{e}
\end{bmatrix}
\end{align*}
\]

\( A \) is Hurwitz if \( K_p \) and \( K_d \) are diagonal and positive-definite \( \rightarrow \lim_{t \to \infty} x(t) = 0 \rightarrow \lim_{t \to \infty} q(t) = q^r(t) \)
Many names for the same approach

This control law:

\[ \tau = M(\dot{\nu}^r - K_d \dot{e} - K_p e) + h \]  

(3)

is known as:

- **Inverse-Dynamics (ID) Control**: because based on inverse dynamics computation.
- **Computed Torque**: because it computes torques needed to get desired accelerations.
- **Feedback Linearization** (from control theory): because it uses state feedback to linearize closed-loop dynamics.

Another variant (with similar properties) exists:

\[ \tau = M(\dot{\nu}^r - K_d \dot{e} - K_p e) + h \]  

(4)
Many names for the same approach

This control law:

\[ \tau = M(\dot{v}^r - K_d \dot{e} - K_p e) + h \]  \hspace{1cm} (3)

is known as:

- **Inverse-Dynamics (ID) Control**: because based on inverse dynamics computation.
- **Computed Torque**: because it computes torques needed to get desired accelerations.
- **Feedback Linearization** (from control theory): because it uses state feedback to linearize closed-loop dynamics.

Another variant (with similar properties) exists:

\[ \tau = M\dot{v}^r - K_d \dot{e} - K_p e + h \]  \hspace{1cm} (4)
Simpler control laws often used for manipulators.

A common option is PD+gravity compensation:

\[
\tau = -K_d \dot{e} - K_p e + \underbrace{g(q)}_{PD \text{ gravity compensation}}
\]  

(5)
Other Control Laws for Manipulators

Simpler control laws often used for manipulators.

A common option is PD + gravity compensation:

\[ \tau = -K_d \dot{e} - K_p e + \underbrace{g(q)}_{\text{PD}} + \underbrace{\int_0^t K_i e(s) \, ds}_{\text{gravity compensation}} \] (5)

Another (even simpler) option is PID control:

\[ \tau = -K_d \dot{e} - K_p e - \int_0^t K_i e(s) \, ds \] (6)

where integral replaces gravity compensation.
Other Control Laws for Manipulators

Simpler control laws often used for manipulators.

A common option is PD + gravity compensation:

\[ \tau = -K_d \dot{e} - K_p e + g(q) \]

(5)

Another (even simpler) option is PID control:

\[ \tau = -K_d \dot{e} - K_p e - \int_0^t K_i e(s) ds \]

(6)

where integral replaces gravity compensation.

Both control laws are stable (so \( q \to q^f \)).
Other Control Laws for Manipulators

Simpler control laws often used for manipulators.

A common option is PD+gravity compensation:

\[
\tau = -K_d \dot{e} - K_p e + g(q) \quad \text{(PD)}
\]

where \( g(q) \) is the gravity compensation.

Another (even simpler) option is PID control:

\[
\tau = -K_d \dot{e} - K_p e - \int_0^t K_i e(s) \, ds \quad \text{(PID)}
\]

where integral replaces gravity compensation.

Both control laws are stable (so \( q \to q^r \)).

In theory “ID control” outperforms “PD+gravity”, which outperforms “PID”.


Other Control Laws for Manipulators

Simpler control laws often used for manipulators.

A common option is PD+gravity compensation:

\[ \tau = -K_d \dot{e} - K_p e + g(q) \]  

(5)

Another (even simpler) option is PID control:

\[ \tau = -K_d \dot{e} - K_p e - \int_0^t K_i e(s) \, ds \]  

(6)

where integral replaces gravity compensation.

Both control laws are stable (so \( q \rightarrow q^r \)).

In theory “ID control” outperforms “PD+gravity”, which outperforms “PID”.

In practice the opposite could occur because of model errors.
Inverse Dynamics Control as Optimization Problem
Solution of optimization problem:

\[
(\tau^*, \dot{\nu}^*) = \arg\min_{\tau, \dot{\nu}} \| \dot{\nu} - \dot{\nu}^d \|^2
\]

subject to \( M \dot{\nu} + h = \tau \) 

with \( \dot{\nu}^d = \dot{\nu}^r - K_d \dot{e} - K_p e \)
Inverse Dynamics (ID) Control as Least-Squares Problem

Solution of optimization problem:

\[(\tau^*, \dot{\mathbf{v}}^*) = \arg\min_{\tau, \dot{\mathbf{v}}} \|\dot{\mathbf{v}} - \dot{\mathbf{v}}^d\|^2\]  

subject to \[M\dot{\mathbf{v}} + h = \tau\]  

with \[\dot{\mathbf{v}}^d = \dot{\mathbf{v}}^r - K_d\dot{e} - K_p e,\] is exactly the ID control law:

\[\tau^* = M\dot{\mathbf{v}}^d + h,\]  

Problem (7) is Least-Squares Program/Problem (LSP).
Inverse Dynamics (ID) Control as Least-Squares Problem

Solution of optimization problem:

\[
(\tau^*, \dot{v}^*) = \arg\min_{\tau, \dot{v}} \|\dot{v} - \dot{v}^d\|^2
\]

subject to \( M\ddot{v} + h = \tau \) \hspace{1cm} (7)

with \( \dot{v}^d = \dot{v}^r - K_d \dot{e} - K_p e \), is exactly the ID control law:

\[
\tau^* = M\dot{v}^d + h, 
\] \hspace{1cm} (8)

No advantage in solving (7) to compute (8), but (7) is starting point to solve more complex problems.
Inverse Dynamics (ID) Control as Least-Squares Problem

Solution of optimization problem:

\[
(\tau^*, \dot{v}^*) = \arg \min_{\tau, \dot{v}} \|\dot{v} - \dot{v}^d\|^2
\]

subject to \[ M\dot{v} + h = \tau \]  \hspace{1cm} (7)

with \(\dot{v}^d = \dot{v}^r - K_d \ddot{e} - K_p e\), is exactly the ID control law:

\[
\tau^* = M\dot{v}^d + h, \hspace{1cm} (8)
\]

No advantage in solving (7) to compute (8), but (7) is starting point to solve more complex problems.

Problem (7) is Least-Squares Program/Problem (LSP).
Least-Squares Programs (LSP) have:

- linear equality/inequality constraints \((Ax \leq b, \text{ or } Ax = b)\)
- 2-norm of linear cost function \((||Ax - b||^2)\)
Least-Squares Programs (LSP) have:

- linear equality/inequality constraints ($Ax \leq b$, or $Ax = b$)
- 2-norm of linear cost function ($||Ax - b||^2$)

LSPs are subclass of convex Quadratic Programs (QPs), which have:

- linear equality/inequality constraints ($Ax \leq b$, or $Ax = b$)
- convex quadratic cost function ($x^T H x + h^T x$, with $H \geq 0$)
Least-Squares Programs (LSP) have:

- linear equality/inequality constraints ($Ax \leq b$, or $Ax = b$)
- 2-norm of linear cost function ($||Ax - b||^2$)

LSPs are subclass of convex Quadratic Programs (QPs), which have:

- linear equality/inequality constraints ($Ax \leq b$, or $Ax = b$)
- convex quadratic cost function ($x^T H x + h^T x$, with $H \geq 0$)

LSPs and convex QPs can be solved extremely fast with off-the-shelf softwares.
Least-Squares Programs (LSP) have:

- linear equality/inequality constraints ($Ax \leq b$, or $Ax = b$)
- 2-norm of linear cost function ($||Ax - b||^2$)

LSPs are subclass of convex Quadratic Programs (QPs), which have:

- linear equality/inequality constraints ($Ax \leq b$, or $Ax = b$)
- convex quadratic cost function ($x^T H x + h^T x$, with $H \succeq 0$)

LSPs and convex QPs can be solved extremely fast with off-the-shelf softwares

→ We can solve LSP/QPs inside 1 kHz control loops!
Adding Torque Limits to ID Control

Take the ID control LSP:

\[
\begin{align*}
\text{minimize} \quad & \left\| \dot{v} - \dot{v}^d \right\|^2 \\
\text{subject to} \quad & M\dot{v} + h = \tau
\end{align*}
\]  

(9)
Adding Torque Limits to ID Control

Take the ID control LSP:

$$\minimize_{\tau, \dot{\nu}} \left\| \dot{\nu} - \dot{\nu}^d \right\|^2$$

subject to

$$M \ddot{\nu} + h = \tau$$ \hspace{1cm} (9)

LSPs allow for linear inequality constraints $\rightarrow$ we can add torque limits:

$$\minimize_{\tau, \dot{\nu}} \left\| \dot{\nu} - \dot{\nu}^d \right\|^2$$

subject to

$$M \ddot{\nu} + h = \tau$$ \hspace{1cm} (10)

$$\tau^\text{min} \leq \tau \leq \tau^\text{max}$$
Adding Torque Limits to ID Control

Take the ID control LSP:

$$\begin{align*}
\text{minimize} & \quad \|\dot{v} - \dot{v}^d\|^2 \\
\text{subject to} & \quad M\dot{v} + h = \tau
\end{align*}$$  \quad (9)$$

LSPs allow for linear inequality constraints → we can add torque limits:

$$\begin{align*}
\text{minimize} & \quad \|\dot{v} - \dot{v}^d\|^2 \\
\text{subject to} & \quad M\dot{v} + h = \tau \\
& \quad \tau_{\text{min}} \leq \tau \leq \tau_{\text{max}}
\end{align*}$$  \quad (10)$$

Main advantage of optimization: inequality constraints.
In electric motors current $i$ is proportional to torque $\tau$:

$$i = k_{\tau} \tau$$ (11)
In electric motors current $i$ is proportional to torque $\tau$:

$$i = k_\tau \tau$$  \hspace{1cm} (11)

Add current limits:

$$\begin{align*}
\text{minimize} & \quad ||\dot{\tau} - \dot{\tau}^d||^2 \\
\text{subject to} & \quad M \dot{\tau} + h = \tau \\
& \quad \tau^{\text{min}} \leq \tau \leq \tau^{\text{max}} \\
& \quad i^{\text{min}} \leq k_\tau \tau \leq i^{\text{max}}
\end{align*}$$  \hspace{1cm} (12)
Adding Joint Velocity Limits

Assuming constant accelerations $\dot{v}$ during time step $\Delta t$:

$$v(t + \Delta t) = v(t) + \Delta t \dot{v}$$  \hspace{1cm} (13)
Adding Joint Velocity Limits

Assuming constant accelerations $\dot{v}$ during time step $\Delta t$:

$$v(t + \Delta t) = v(t) + \Delta t \dot{v}$$  \hspace{1cm} (13)

Add joint velocity limits:

$$\begin{align*}
\text{minimize} & \quad \| \dot{v} - \dot{v}^d \|^2 \\
\text{subject to} & \quad M \ddot{v} + h = \tau \\
& \quad \tau^{\min} \leq \tau \leq \tau^{\max} \\
& \quad i^{\min} \leq k_\tau \tau \leq i^{\max} \\
& \quad v^{\min} \leq v + \Delta t \dot{v} \leq v^{\max}
\end{align*}$$  \hspace{1cm} (14)
Adding Joint Position Limits

Could use same trick for position limits:

\[ q(t + \Delta t) = q(t) + \Delta t \nu(t) + \frac{1}{2} \Delta t^2 \dot{\nu} \]  

(15)

However, this can result in high accelerations, typically incompatible with torque/current limits → unfeasible LSP.

Better approaches exist [1, 8, 2], but we don't discuss them here.
Could use same trick for position limits:

\[ q(t + \Delta t) = q(t) + \Delta t v(t) + \frac{1}{2} \Delta t^2 \dot{v} \] (15)

However, this can result in high accelerations, typically incompatible with torque/current limits → unfeasible LSP.
Can use same trick for position limits:

\[ q(t + \Delta t) = q(t) + \Delta t \, v(t) + \frac{1}{2} \Delta t^2 \dot{v} \]  

(15)

However, this can result in high accelerations, typically incompatible with torque/current limits → unfeasible LSP.

Better approaches exist [1, 8, 2], but we don’t discuss them here.
Inverse-Dynamics Control: \[ \tau = M(\dot{v}^r - K_d \dot{e} - K_p e) + h \]
Summary

Inverse-Dynamics Control: \[ \tau = M(\dot{\nu}^r - K_d \dot{e} - K_p e) + h \]

Other version: \[ \tau = M\dot{\nu}^r - K_d \dot{e} - K_p e + h \]
Inverse-Dynamics Control:  \[ \tau = M(\ddot{v}^r - K_d \dot{e} - K_p e) + h \]

Other version:  \[ \tau = M\ddot{v}^r - K_d \dot{e} - K_p e + h \]

PD + gravity compensation:  \[ \tau = -K_d \dot{e} - K_p e + g(q) \]
Inverse-Dynamics Control: $\tau = M(\ddot{v}^r - K_d \dot{e} - K_p e) + h$

Other version: $\tau = M\ddot{v}^r - K_d \dot{e} - K_p e + h$

PD + gravity compensation: $\tau = -K_d \dot{e} - K_p e + g(q)$

PID: $\tau = -K_d \dot{e} - K_p e - \int_0^t K_i e(s)ds$
Summary

**Inverse-Dynamics Control:**
\[ \tau = M(\dot{v}^r - K_d \dot{e} - K_p e) + h \]

Other version:
\[ \tau = M\dot{v}^r - K_d \dot{e} - K_p e + h \]

PD + gravity compensation:
\[ \tau = -K_d \dot{e} - K_p e + g(q) \]

PID:
\[ \tau = -K_d \dot{e} - K_p e - \int_0^t K_i e(s) ds \]

**ID Control as LSP:**
\[
\begin{align*}
\text{minimize} & \quad ||\dot{v} - \dot{v}^d||^2 \\
\text{subject to} & \quad M\dot{v} + h = \tau
\end{align*}
\]
Summary

**Inverse-Dynamics Control:**  
\[ \tau = M(\dot{v}^r - K_d \dot{e} - K_p e) + h \]

Other version:  
\[ \tau = M\dot{v}^r - K_d \dot{e} - K_p e + h \]

PD + gravity compensation:  
\[ \tau = -K_d \dot{e} - K_p e + g(q) \]

PID:  
\[ \tau = -K_d \dot{e} - K_p e - \int_0^t K_i e(s)ds \]

**ID Control as LSP:**

minimize  
\[ \|\dot{v} - \dot{v}^d\|^2 \]

subject to  
\[ M\dot{v} + h = \tau \]
\[ \tau_{\text{min}} \leq \tau \leq \tau_{\text{max}} \]
Summary

**Inverse-Dynamics Control:**
\[ \tau = M(\dot{v}^r - K_d \dot{e} - K_p e) + h \]

Other version:
\[ \tau = M\ddot{v}^r - K_d \dot{e} - K_p e + h \]

PD + gravity compensation:
\[ \tau = -K_d \dot{e} - K_p e + g(q) \]

PID:
\[ \tau = -K_d \dot{e} - K_p e - \int_0^t K_i e(s) ds \]

**ID Control as LSP:**

\[
\begin{align*}
\text{minimize} & \quad \| \dot{v} - \dot{v}^d \|^2 \\
\text{subject to} & \quad M\ddot{v} + h = \tau \\
& \quad \tau^{\text{min}} \leq \tau \leq \tau^{\text{max}} \\
& \quad \dot{i}^{\text{min}} \leq k_T \tau \leq \dot{i}^{\text{max}}
\end{align*}
\]
Summary

**Inverse-Dynamics Control:**
\[ \tau = M(\dot{v}^r - K_d \dot{e} - K_p e) + h \]

Other version:
\[ \tau = M\dot{v}^r - K_d \dot{e} - K_p e + h \]

PD + gravity compensation:
\[ \tau = -K_d \dot{e} - K_p e + g(q) \]

PID:
\[ \tau = -K_d \dot{e} - K_p e - \int_0^t K_i e(s) ds \]

**ID Control as LSP:**

\[
\begin{align*}
\text{minimize} & \quad \| \dot{v} - \dot{v}^d \|^2 \\
\text{subject to} & \quad M\dot{v} + h = \tau \\
& \quad \tau^{min} \leq \tau \leq \tau^{max} \\
& \quad i^{min} \leq k_T \tau \leq i^{max} \\
& \quad v^{min} \leq v + \Delta t \dot{v} \leq v^{max}
\end{align*}
\]


Hierarchical Quadratic Programming: Fast Online Humanoid-Robot Motion Generation.  

O. Khatib.  
A unified approach for motion and force control of robot manipulators: The operational space formulation.  

M. Mistry, J. Buchli, and S. Schaal.  
Inverse dynamics control of floating base systems using orthogonal decomposition.  


L. Sentis and O. Khatib.

Synthesis of whole-body behaviors through hierarchical control of behavioral primitives.