

B Arbeiten mit ManDy

B1 ManDy: Einheitliche Programmier- Simulations- und Visualisierungsumgebung (Übersicht)

Das Programmsystem **ManDy** (**Manipulator Dynamics**) wurde an der Hochschule Darmstadt im Zentrum für Robotik als eine einheitliche Programmier-, Simulations- und Visualisierungsumgebung für Mehrachskinematiken wie Roboter- und Manipulatorarme oder anderer Mehrachsmaschinen entwickelt. Der typische Ablauf eines Auslegungsprozesses mit Mandy ist in Bild B1 dargestellt.

Der Anwender kann **beliebige Kinematiken** mit offener kinematischer Kette über eine einfache grafische Schnittstelle definieren, wobei die maximale Anzahl der Gelenke 10 beträgt. Die Eingabe der kinematischen Struktur erfolgt menuegeführt auf der Basis der in der Robotertechnik üblichen Denavit-Hartenberg-Konvention (s. Kap 2.2).

ManDy stellt eine Oberfläche zur Off-line-Programmierung von Bewegungsbefehlen zur Verfügung. Die Befehle werden automatisch an die definierte Kinematik angepasst. Es besteht außerdem die Möglichkeit, die programmierte Bewegung mit einem **dreidimensionalen Animationsmodell** darstellen zu lassen. Das 3D-Modell wird in VRML automatisch aus Matlab heraus erzeugt und in einem Web-Browser zur Anzeige gebracht. Die **Bewegungsfähigkeit des Armes** und die **Erreichbarkeit verschiedener Zielstellungen** kann auf diese Weise untersucht werden. Bei unerreichbaren Zielstellungen können die Kinematikparameter des betrachteten Roboters solange angepasst werden, bis zufrieden stellende Ergebnisse erzielt werden (s. Bild B1)

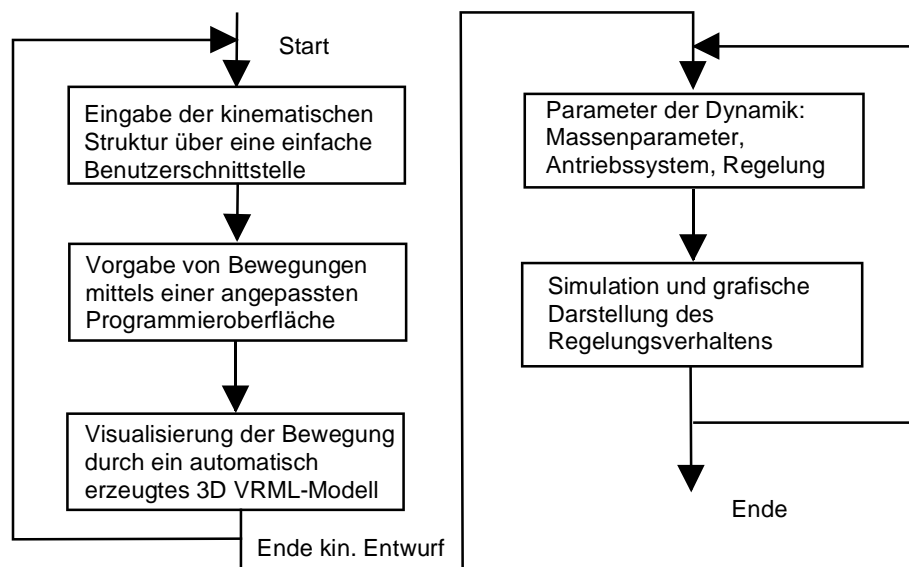


Bild B1 Übersicht zu ManDy

Als nächstes kann geprüft werden, ob die geforderte Bewegung bei dem notwendigen mechanischen Aufbau, zur Verfügung stehenden Antriebssystemen und der eingesetzten Regelung überhaupt erreicht werden kann. Das Modell des Armes wird daher durch die **Dynamikparameter** erweitert (Massen, Trägheiten und Schwerpunkte der Armteile sowie Angaben zu Motoren und Getrieben). Der Anwender kann vier verschiedene **Regelungsstrukturen** auswählen und parametrisieren. Die simulierten Bewegungen können wieder durch eine 3D-Animation mit den Sollbewegungen verglichen werden. Alle Regelungs- und Stellgrößen können auch durch an die Gelenkzahl angepasste grafische Darstellungen bewertet werden. Ist die Regelungsgüte nicht zufrieden stellend, besteht die Möglichkeit die Regelung zu modifizieren und/oder die Parameter der Dynamik zu verändern (Bild B1).

B2 Installation von ManDy

Die Programmdateien und die Parameterdateien von ManDy müssen zunächst in einen beliebigen Ordner auf der Festplatte kopiert werden (z. B. `C:\mandy\`).

Nach dem Start von Matlab muss in das Verzeichnis von ManDy gewechselt werden (`cd c:\mandy`). Dort kann das Programm mit dem Befehl `mandy` gestartet werden. Es erscheint das Startfenster (Bild B2).

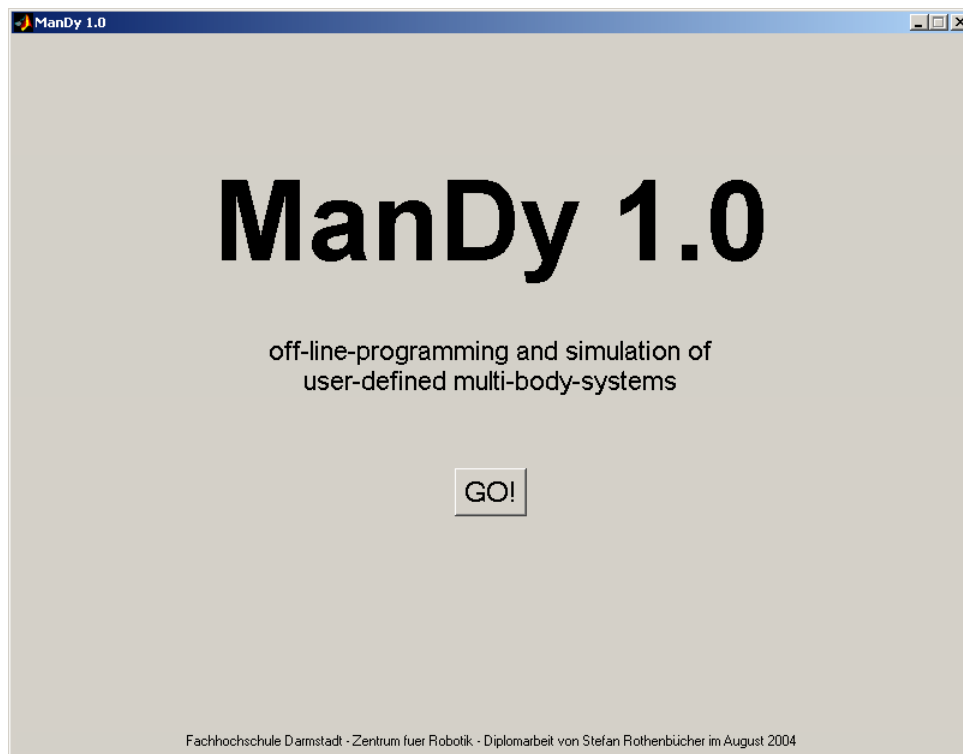


Bild B2 Startmenü von ManDy

Nach einem Klick auf den GO!-Button öffnet sich die Off-line-Programmierungsumgebung. Es wird zur Auswahl eines Roboters aufgefordert.

Beim ersten Start von ManDy sollten die Pfade eingestellt werden. Unter dem Menüpunkt *options* muss der Pfad zu den Parameterdateien angegeben werden. Im Normalfall ist dies derselbe Pfad wie der Installationspfad von ManDy (z. B. *C:\mandy*). Die zweite Einstellung im Optionenmenü, ist der Pfad zu der ausführbaren Datei des VRML-fähigen Browsers. Voraussetzung für die Darstellung des VRML-Modells ist ein installierter VRML-fähiger Web-Browser (z. B. *C:\Programme\Mozilla Firefox\firefox.exe*).

Für die gängigen Web-Browser, wie Mozilla Firefox und Microsoft Internet-Explorer, können verschiedene kostenlose Plug-Ins aus dem Internet heruntergeladen werden. Die verbreitetsten Plug-Ins sind:

Cortona VRML Client 4.2 von *ParallelGraphics*¹

Cosmo Player 2.1 von *CosmoSoftware*²

B3 Definition und Modifikation von Robotern

In ManDy besteht die Möglichkeit neue Roboter mit offener kinematischer Kette zu definieren oder schon definierte Roboter zu modifizieren.

Wählt man in der Menüleiste den Menüpunkt *robot* → *create new* aus, öffnet sich der Eingabedialog für die Kinematikparameter des Roboters (Bild B3):

¹ <http://www.parallelgraphics.com/products/cortona/>

² <http://ca.com/cosmo/>

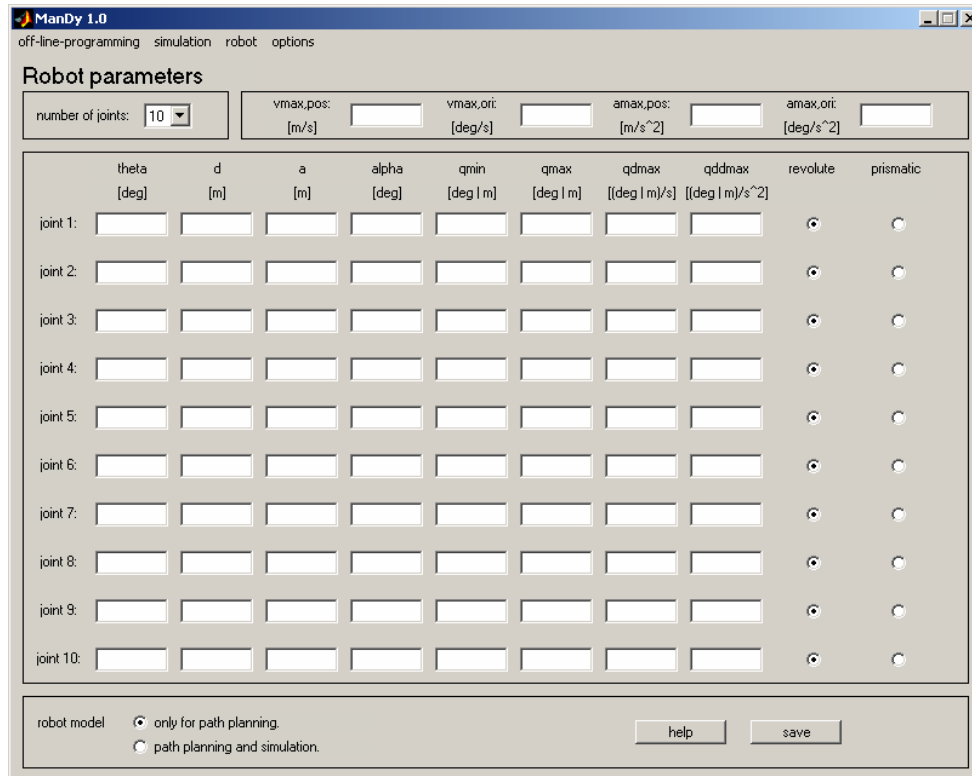


Bild B3 Parameter der Kinematik

Zunächst muss im Popup-Menü die Anzahl der Robotergerlenke (*number of joints*) ausgewählt werden. Die maximale Anzahl von Gelenken beträgt in ManDy 10. Die Anzahl der Eingabefelder für die Roboterparameter wird daraufhin automatisch an die ausgewählte Gelenkanzahl angepasst.

Rechts neben der Auswahl der Gelenkanzahl befinden sich vier Eingabefelder. Hier müssen die maximalen kartesischen Geschwindigkeiten und Beschleunigungen angegeben werden. Im ersten Feld ($v_{max, pos}$) wird die maximale Bahngeschwindigkeit des TCP in m/s, im zweiten Feld ($v_{max, ori}$) die maximale Orientierungsgeschwindigkeit in °/s eingetragen. Die beiden letzten Felder müssen die maximalen Bahnbeschleunigung ($a_{max, pos}$) in m/s^2 und die maximale Orientierungsbeschleunigung ($a_{max, ori}$) in $°/s^2$ beinhalten.

Der nächste Schritt ist die Auswahl der Gelenkart. Auf der rechten Seite kann über Radio-Buttons für jedes Gelenk ausgewählt werden, ob es ein rotatorisches (*revolute*) oder ein translatorisches (*prismatic*) Gelenk sein soll.

Danach werden die Kinematikparameter des Roboters eingetragen. In der Tabelle B1 werden die einzelnen Parameter erläutert:

Tabelle B1: Parameter der Kinematik

Parameter	Beschreibung	Einheit
<i>theta</i>	DENAVIT-HARTENBERG-Parameter θ	°
<i>d</i>	DENAVIT-HARTENBERG-Parameter <i>d</i>	m
<i>a</i>	DENAVIT-HARTENBERG-Parameter <i>a</i>	m
<i>alpha</i>	DENAVIT-HARTENBERG-Parameter α	°
<i>qmin</i>	untere Gelenkkoordinatenbegrenzung	° oder m
<i>qmax</i>	obere Gelenkkoordinatenbegrenzung	° oder m
<i>qdmx</i>	maximale Gelenkgeschwindigkeit	°/s oder m/s
<i>qddmx</i>	maximale Gelenkbeschleunigung	°/s ² oder m/s ²

Die Einheiten der Parameter richten sich nach der Art des Gelenks. Die unteren und oberen Gelenkkoordinatenbegrenzungen müssen bei rotatorischen Gelenken beispielsweise in Grad, bei translatorischen Gelenken in Metern angegeben werden.

Im unteren Bereich des Menues (s. Bild B3) können weitere Einstellungen vorgenommen werden. Über Radio-Buttons kann ausgewählt werden, ob das Robotermodell, welches gerade eingegeben wird, nur für die Bahnplanung (*only for path planning*) oder für Bahnplanung und Simulation des Regelungsverhaltens (*path planning and simulation*) verwendet werden soll. Für den ersten Fall (Bahnplanung) reichen die auf dieser Seite eingegebenen Roboterparameter aus. In der Off-line-Programmierungsumgebung können dann Roboterprogramme erstellt und übersetzt werden. Weiterhin kann die Visualisierung der Raumfahrt erfolgen. Das Speichern des Robotermodells erfolgt über den *save*-Button. Es öffnet sich ein Dialog, in dem ein Pfad und ein Dateiname ausgewählt werden müssen. Als Dateinamen gibt man am besten den Namen des Roboters, z. B. *scara* oder *reis_rv6*, an. Der standardmäßig vorgegebene Pfad `...\mandy\robots` sollte beibehalten werden, da bei der späteren Auswahl des Roboters in der Off-line-Programmierungsumgebung wieder standardmäßig dieser Pfad vorgeschlagen wird. Es ist noch zu beachten, dass die eingegebenen Parameter nicht auf ihre Richtigkeit überprüft werden und dass leere Eingabefelder vor dem Speichern mit Nullen gefüllt werden.

Soll der Roboter für Bahnplanung und Simulation des Regelungsverhaltens verwendet werden, so muss der zweite Radio-Button (*path planning and simulation*) ausgewählt werden. Nach der Auswahl erscheint unten rechts ein neuer Button *next >>*. Klickt man auf diesen Button, erscheint eine weitere Seite mit neuen Eingabefeldern, in die die Dynamikparameter des Roboters, die für die Simulation des Bewegungsverhaltens benötigt werden, eingetragen werden müssen. In Bild B4 ist das Eingabefenster für die weiteren Parameter dargestellt:

ManDy 1.0
off-line-programming simulation robot options

Robot parameters

gravity vector: g_x [m/s²] g_y [m/s²] g_z [m/s²]

	m [kg]	C [Nm/A]	JA [kg·m ²]	FD [Nm ² /deg]	FM [Nm ² /deg]	lAmax [A]	K [Nm/deg]	D [Nm ² /deg]
joint 1:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 2:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 3:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 4:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 5:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 6:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 7:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 8:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 9:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 10:	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

<< back help SBVB next >>

Bild B4 Parameter der Dynamik, Teil 1

Auf dieser Seite muss zunächst der Vektor der Gravitationsbeschleunigung, der sich auf das Weltkoordinatensystem K_0 bezieht, angegeben werden. In die drei Eingabefelder werden die x -, y - und z -Komponente des Vektors eingetragen. Zeigt die z_0 -Achse beispielsweise nach oben, so muss in das Eingabefeld für die z -Komponente des Gravitationsvektors der Wert -9.81 , in die beiden anderen Felder jeweils eine 0 eingetragen werden. In die Eingabefelder darunter müssen die in Tabelle Tab. B2 angegebenen Parameter eingetragen werden.

Tabelle B2: Parameter der Dynamik, Teil 1

Parameter	Beschreibung	Einheit
m	Armteilmasse	kg
C	Motorkonstante	N·m/A
JA	Ankerträgheiten	Kg·m ²
FD	Reibungskoeffizient für geschwindigkeitsabhängige abtriebsseitige Reibung	N·m·s/° oder N·s/m
FM	Reibungskoeffizient für geschwindigkeitsabhängige antriebsseitige Reibung	N·m·s/° oder N·s/m
I_{Amax}	maximale Ankerströme	A
K	Steifigkeit	N·m/° oder kg/s
D	Dämpfung	N·m·s/° oder kg/s

Die Einheiten der Parameter richten sich auch hier teilweise nach der Art des Gelenks (rotatorisch oder translatorisch). Sind in Tabelle B2 zwei Einheiten angegeben, so bezieht sich die erste Einheit immer auf rotatorische Gelenke, die zweite Einheit auf translatorische Gelenke.

Mit einem weiteren Klick auf den *next*-Button erscheint eine weitere Seite mit Eingabefeldern für Roboterparameter (Bild B5).

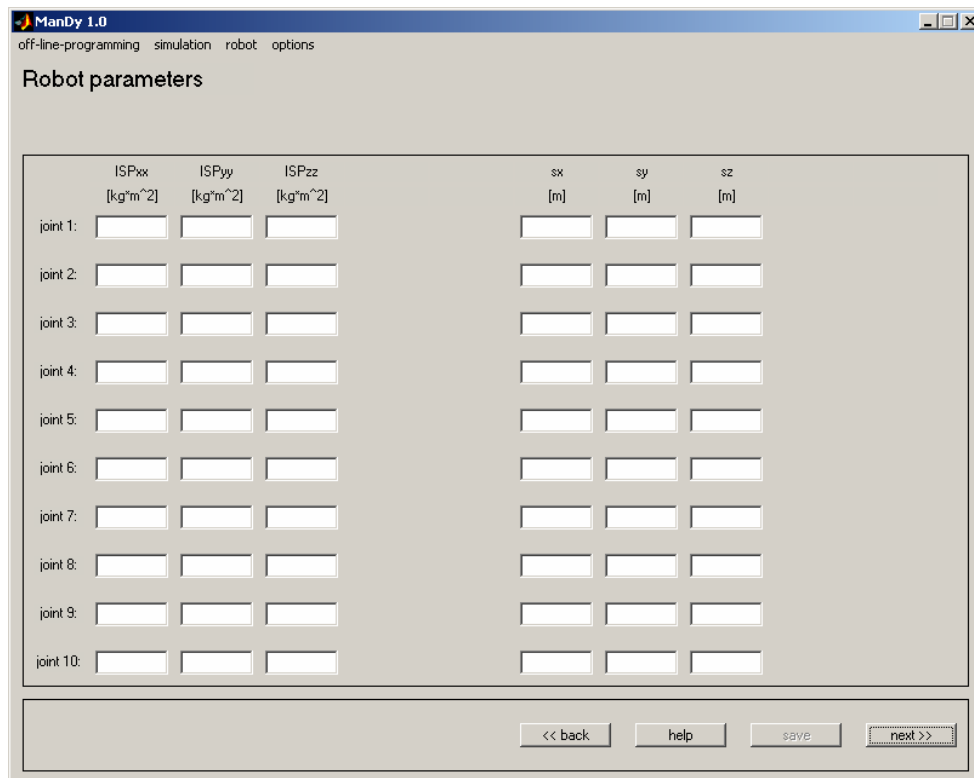


Bild B5 Parameter der Dynamik, Teil 2

Hier müssen die Trägheitstensoren der Armteile und die Schwerpunktsvektoren angegeben werden. Von den (3·3)-Trägheitstensoren für jedes Armteil wird jeweils nur die Hauptdiagonale (ISP_{xx} , ISP_{yy} und ISP_{zz}) abgefragt. Nebendiagonalen werden aufgrund ihres relativ geringen Einflusses vernachlässigt. Der Trägheitstensor wird jeweils auf das armteileigene Koordinatensystem bezogen. Die Einheiten der einzelnen Komponenten sind $\text{kg}\cdot\text{m}^2$. Der Schwerpunktsvektor beschreibt die Lage des Schwerpunktes ebenfalls bezüglich des armteileigenen Koordinatensystems (Einheit: m).

Bei einem weiteren Klick auf den *next*-Button erscheint die letzte Seite (Bild B6). Hier ist das Antriebssystem zu spezifizieren. Zuerst wird die Getriebematrix (*gear matrix*) S_G eingetragen (s. auch Gln. 6.29, 6.30, 6.32). Es stehen dazu $N\cdot N$ Eingabefelder zur Verfügung (N entspricht der Anzahl der Gelenke). In die Hauptdiagonale werden die Übersetzungen eingetragen. Getriebekopplungen können durch einen Eintrag in die entsprechenden Nebendiagonalen berücksichtigt werden. Nebendiagonalelemente, die nicht eingetragen werden, werden von ManDy beim Abspeichern automatisch mit Nullen gefüllt.

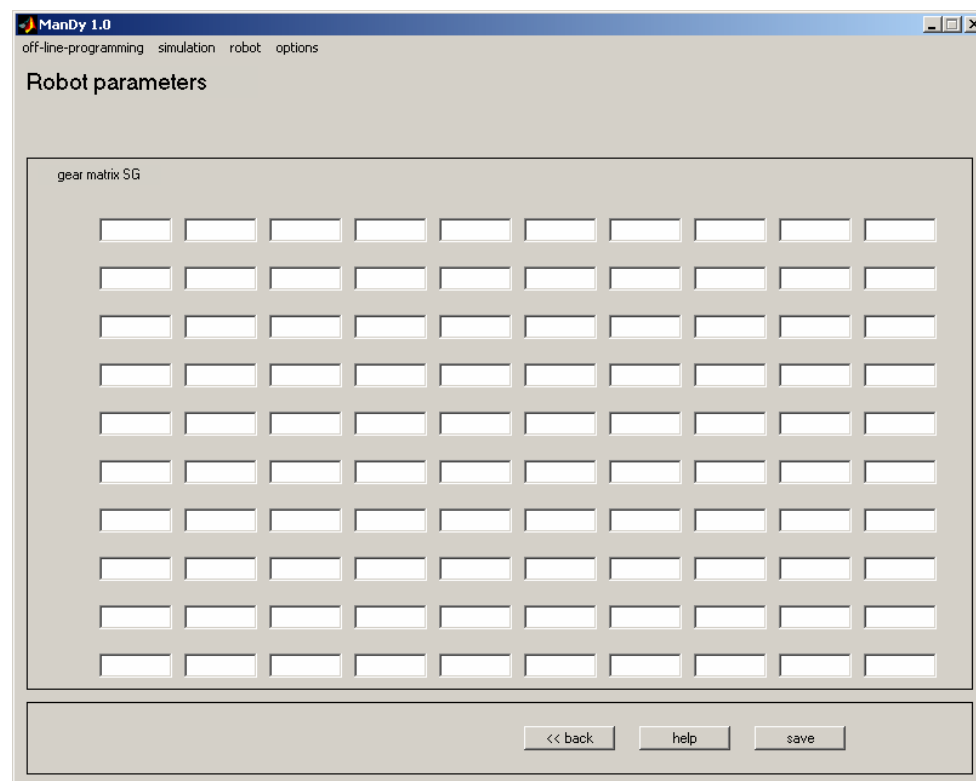


Bild B6 Getriebematrix S_G

Bevor die Roboterparameter eines Fensters abgespeichert werden, kann über die *back*- und *next*-Buttons nochmals überprüft werden, ob die eingegebenen Parameter korrekt sind.

Zum Abspeichern aller Roboterparameter muss auf die letzte Seite gewechselt werden. Dort befindet sich der *save*-Button, mit dem sämtliche Roboterparameter abgespeichert werden können.

Wählt man im Menü *robot* → *edit existent* öffnet sich ein Dialog, in dem zur Auswahl einer Parameterdatei eines Roboters aufgefordert wird. Nach der Auswahl eines Robotermodells öffnet sich das Fenster, wie es in Bild B3 dargestellt ist. In den Eingabefeldern befinden sich jetzt aber bereits die Parameter des aufgerufenen Roboters. Nachdem die gewünschten Änderungen durchgeführt wurden, können die neuen Parameter, mit Hilfe des *save*-Buttons, entweder unter dem gleichen Dateinamen wie vorher oder unter einem neuen Dateinamen gespeichert werden. Der *save*-Button befindet sich wieder auf der ersten oder letzten Parameterseite, je nach dem, ob der Roboter nur für die Bahnplanung oder für Bahnplanung und Simulation des Regelungsverhaltens eingesetzt werden soll.

B3 Off-line-Programmierung

Die Off-line-Programmierung von ManDy (s. Bild B7) besteht aus den drei folgenden Hauptkomponenten:

1. Erstellung und Verwaltung von Roboterprogrammen
2. Übersetzung von Roboterprogrammen im Interpreter
3. Visualisierung von programmierten Bahnen

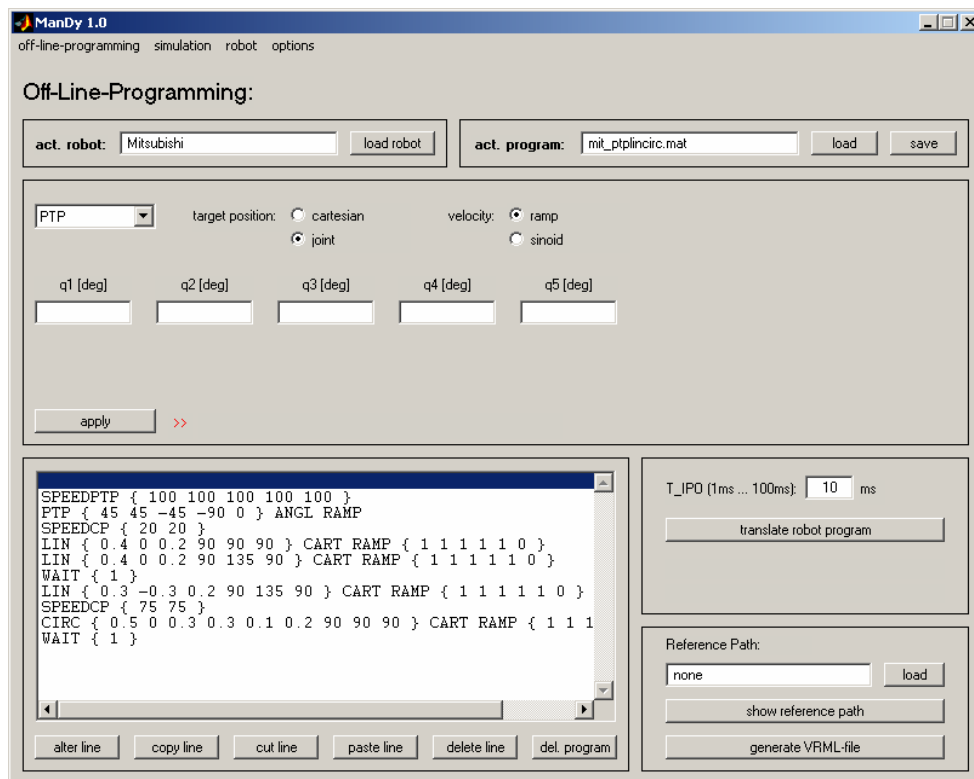


Bild B7 Fenster für die Off-line Programmierung

Beim Start der Off-line-Programmierungsumgebung erscheint ein Dialog, in dem die Parameterdatei des Roboters ausgewählt werden muss, für den man ein Roboterprogramm erstellen möchte. Der Roboter, der gerade verwendet wird, wird in einem Feld links oben angezeigt. In Bild B7 ist dies z. B. ein *Mitsubishi*-Roboter. Rechts neben diesem Feld befindet sich der *load robot*-Button. Mit Hilfe dieses Buttons kann eine Parameterdatei für einen anderen Roboter geladen werden.

Über den *load*- bzw. *save*-Button im rechten oberen Teil des Fensters kann ein existierendes Roboterprogramm geladen werden bzw. ein gerade erstelltes Roboterprogramm gespeichert werden. Im Feld links neben den Buttons wird der Name des Roboterprogramms angezeigt, das gerade bearbeitet wird. Beim Laden eines Roboterprogramms ist darauf zu achten, dass das Roboterprogramm auch zu dem aktuell verwendeten Roboter gehört. Ist dies nicht der Fall erscheint eine Fehlermeldung.

Im mittleren Teil des Fensters (Bild B7) kann ein neuer Roboterbefehl ausgewählt und parametrisiert werden. Die Auswahl des gewünschten Roboterbefehls erfolgt über ein Popup-Menü. Die Tabelle B3 enthält eine Liste aller verfügbaren Roboterbefehle. Nach

der Auswahl eines Befehls wird die Eingabeoberfläche für die Befehlsparameter automatisch an den gewählten Roboterbefehl angepasst. Nach der Parametrisierung des Roboterbefehls kann der Befehl mit Hilfe des *apply*-Buttons in das Roboterprogramm übernommen werden. Vorher wird jedoch noch eine Überprüfung der Angaben durchgeführt, damit es bei der späteren Übersetzung des Roboterprogramms nicht Fehlern kommt. Es wird beispielsweise geprüft, ob in jedem Eingabefeld eine Angabe gemacht wurde und ob die Angaben als Zahlen interpretiert werden können. Wenn eine Unstimmigkeit gefunden wird, dann erscheint rechts neben dem *apply*-Button eine Fehlermeldung und der Roboterbefehl wird nicht in das Roboterprogramm übernommen.

Die Befehle, des gerade in Bearbeitung befindliche Roboterprogramms werden in einem Fenster links unten angezeigt (Bild B7). Unter diesem Fenster befinden sich weitere Buttons, mit denen das Roboterprogramm editiert werden kann und im Folgenden beschrieben werden. Mit dem *alter line*-Button kann die aktuell im Roboterprogramm markierte (blau hervorgehobene) Zeile verändert werden. Die Befehlsparameter werden in die Eingabefelder für die Parametrisierung der Roboterbefehle geschrieben und können dort abgeändert werden. Über den *apply*-Button kann der abgeänderte Befehl wieder in das Roboterprogramm übernommen werden. Der aktuelle Roboterbefehl wird immer nach der blau hervorgehobenen Zeile im Roboterprogramm eingefügt. Um einen Befehl auch noch später an den Programmbeginn schreiben zu können, ist die erste Zeile des Programmfensters immer eine Leerzeile. Markiert man diese Zeile, wird der aktuelle Roboterbefehl durch Betätigung des *apply*-Buttons danach eingefügt und steht somit am Programmanfang. Die erste Zeile des Programmfensters ist kein Teil des Roboterprogramms und kann deshalb auch nicht gelöscht werden. Mit dem *copy*-Button (kopieren) wird die gerade markierte Zeile des Roboterprogramms in die Zwischenablage kopiert. Mit dem *cut*-Button (ausschneiden) wird die gerade markierte Zeile des Roboterprogramms ebenfalls in die Zwischenablage kopiert. Weiterhin wird die Zeile aus dem Roboterprogramm entfernt. Über den *paste*-Button (einfügen) kann eine Zeile des Roboterprogramms, die über den *copy*- oder *cut*-Button in die Zwischenablage übernommen wurde, an einer beliebigen Stelle des Roboterprogramms eingefügt werden. Mit Hilfe des *delete line*-Button wird die gerade markierte Zeile des Roboterprogramms gelöscht. Mit dem *del. program*-Button kann das gesamte Roboterprogramm gelöscht werden.

Tabelle B3: Programmierbefehle

Befehl	Parameter	Beschreibung
<i>PTP</i>	q_1, \dots, q_N oder x, y, z, A, B, C	Point-To-Point-Bahn: Es wird von der aktuellen Stellung zur in Gelenkkoordinaten oder kartesischen Koordinaten angegebenen Zielstellung verfahren. Es kann ein rampen- oder sinoidenförmiges Geschwindigkeitsprofil ausgewählt werden.
<i>PTPREL</i>	$\Delta q_1, \dots, \Delta q_N$ oder $\Delta x, \Delta y, \Delta z,$ $\Delta A, \Delta B, \Delta C$	relative Point-To-Point-Bahn: Es wird von der aktuellen Stellung zur in relativen Gelenkkoordinaten oder relativen kartesischen Koordinaten angegebenen Zielstellung verfahren. Es kann ein rampen- oder sinoidenförmiges Geschwindigkeitsprofil ausgewählt werden.
<i>SPEEDPTP</i>	v_1, \dots, v_N in %	Gelenkgeschwindigkeitsvorgabe für PTP-Bahnen:

		Definition der Gelenkgeschwindigkeiten in Prozent der max. Gelenkgeschwindigkeiten. Die Gelenkgeschwindigkeitsvorgabe gilt solange für alle PTP-Bahnen, bis eine neue Vorgabe erfolgt. Voreinstellung: 50 %
<i>ACCELPTP</i>	a_1, \dots, a_N in %	Gelenkbeschleunigungsvorgabe für PTP-Bahnen: Definition der Gelenkbeschleunigungen in Prozent der max. Gelenkbeschleunigungen. Die Gelenkbeschleunigungsvorgabe gilt solange für alle PTP-Bahnen, bis eine neue Vorgabe erfolgt. Voreinstellung: 50 %
<i>LIN</i>	q_1, \dots, q_N oder $x, y, z,$ $A, B,$ C	Linearbahn: Der TCP fährt von der aktuellen Stellung zu der in Gelenkkoordinaten oder kartesischen Koordinaten angegebenen Zielstellung auf einer Geraden. Die EULER -Winkel werden kontinuierlich von den aktuellen Werten zu den Zielwerten nachgestellt. Es kann ein rampen- oder sinoidenförmiges Geschwindigkeitsprofil ausgewählt werden.
<i>LINREL</i>	$\Delta q_1, \dots, \Delta q_N$ oder $\Delta x, \Delta y, \Delta z,$ $\Delta A,$ $\Delta B,$ ΔC	Relative Linearbahn: Der TCP fährt von der aktuellen Stellung zu der in relativen Gelenkkoordinaten oder relativen kartesischen Koordinaten angegebenen Zielstellung auf einer Geraden. Die EULER -Winkel werden kontinuierlich von den aktuellen Werten zu den relativ angegebenen Zielwerten nachgestellt. Es kann ein rampen- oder sinoidenförmiges Geschwindigkeitsprofil ausgewählt werden.
<i>CIRC</i>	x_H, y_H, z_H und x, y, z, A, B, C	Zirkularbahn: Der TCP fährt auf einem Kreisbogen von der aktuellen Stellung über den Hilfspunkt (x_H, y_H, z_H) zur Zielstellung. Die EULER -Winkel werden kontinuierlich von den aktuellen Werten zu den Zielwerten nachgestellt. Es kann ein rampen- oder sinoidenförmiges Geschwindigkeitsprofil ausgewählt werden.
<i>CIRCREL</i>	$\Delta x_H, \Delta y_H, \Delta z_H$ und $\Delta x, \Delta y, \Delta z,$ $\Delta A, \Delta B, \Delta C$	Relative Zirkularbahn: Der TCP fährt auf einem Kreisbogen von der aktuellen Stellung über den relativ angegebenen Hilfspunkt zum in relativ angegebenen Zielpunkt. Die EULER -Winkel werden kontinuierlich von den aktuellen Werten zu den relativ angegebenen Zielwerten nachgestellt. Es kann ein rampen- oder sinoidenförmiges Geschwindigkeitsprofil ausgewählt werden.
<i>SPEEDCP</i>	v_{pos} und v_{ori} in %	Geschwindigkeitsvorgabe für CP-Bahnen: Vorgabe der Bahngeschwindigkeit und der Orientierungsgeschwindigkeit in Prozent der Maximalwerte. Die Geschwindigkeitsvorgaben gelten solange für alle CP-Bahnen, bis eine neue Vorgabe erfolgt.

		Voreinstellung: 50 %
ACCELCP	a_{pos} und a_{ori} in %	Beschleunigungsvorgabe für CP-Bahnen: Vorgabe der Bahnbeschleunigung und der Orientierungsbeschleunigung in Prozent der Maximalwerte. Die Beschleunigungsvorgaben gelten solange für alle CP-Bahnen, bis eine neue Vorgabe erfolgt. Voreinstellung: 50 %
WAIT	t	Warten: Es wird die Zeit t gewartet. Während dieser Zeit werden die Gelenksollwerte konstant gehalten.
COMMENT		Einfügen einer Kommentarzeile

Zu den Relativbewegungen in kartesischen Koordinaten muss noch angemerkt werden, dass sich die Angaben Δx , Δy und Δz immer auf das Weltkoordinatensystem K_0 beziehen. Die relativen EULER-Winkel ΔA , ΔB und ΔC beziehen sich immer auf das Werkzeugkoordinatensystem K_N .

Eine Besonderheit bei den CP-Befehlen (*LIN*, *LINREL*, *CIRC*, *CIRCREL*) ist der so genannte Maskenvektor (*mask-vector*). Die zusätzliche Angabe des (6-1)-Maskenvektors ist bei Robotern mit weniger als sechs Gelenken bzw. Freiheitsgraden erforderlich, damit der Bahnplanungsalgorithmus die Freiheitsgrade des Roboters kennt. Der Maskenvektor hat die Form $\mathbf{m} = (m_x, m_y, m_z, m_C, m_B, m_A)$. Die Elemente des Vektors können die Werte 0 oder 1 annehmen und definieren welche Freiheitsgrade vorhanden sind. Die ersten drei Elemente stehen für die Translationen in x -, y - und z -Richtung und die letzten drei Elemente für die Rotationen um die x -, y - und z -Achse des Koordinatensystems K_0 . So hat ein Roboter mit drei rotatorischen Gelenken, deren Gelenkachsen alle senkrecht zur x - y -Ebene sind, drei unabhängige Freiheitsgrade, jeweils einen translatorischen Freiheitsgrad in x - und in y -Richtung sowie einen rotatorischen Freiheitsgrad um die z -Achse, also den EULER-Winkel A . Für diesen Roboter müsste der Maskenvektor an der Stelle m_x , m_y und m_A jeweils mit 1 besetzt werden, die Stellen m_z , m_B und m_C jeweils mit 0. Für Roboter mit sechs oder mehr Gelenken ist der Maskenvektor im Allgemeinen voll mit Einsen besetzt. Der Maskenvektor wird bei CP-Befehlen über sechs Radio-Buttons für die Werte m_x , m_y , m_z , m_C , m_B und m_A festgelegt. Für den oben angesprochenen Roboter ist die Parametrisierung eines *LIN*-Befehls in Bild B8 dargestellt. In die Eingabefelder z , A und B können beliebige Werte eingetragen werden. Diese Werte werden bei der Bahnplanung ignoriert.

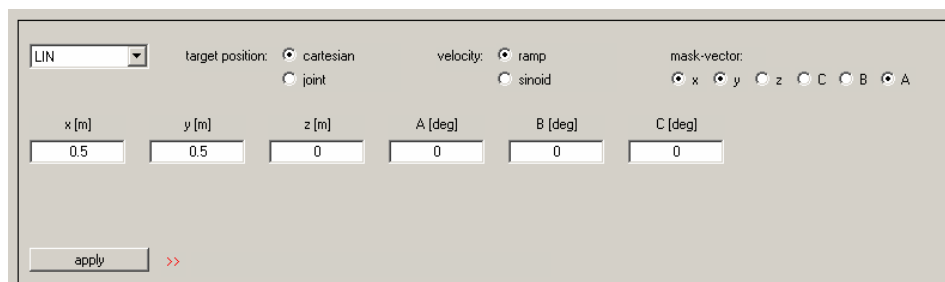


Bild B8 Maskenvektor für Roboter mit weniger als sechs Freiheitsgraden

Nach der Fertigstellung des Roboterprogramms kann dieses übersetzt werden. Das Ergebnis der Übersetzung sind Sollwerte der Gelenkkoordinaten für jeden Interpolationszeitpunkt $t_k = k \cdot T_{ipo}$. Vor der Übersetzung besteht die Möglichkeit, die Interpolationszeit T_{ipo} zu verändern. Dazu kann im Eingabefeld ein Wert zwischen 1 ms und 100 ms eingetragen werden (siehe Bild B9). Der Standardwert beträgt 10 ms. Die Übersetzung kann mit dem Button *translate robot program* gestartet werden. Unter diesem Button befinden sich zwei Textfelder. Im oberen Textfeld werden evtl. auftretende Fehler, Warnungen etc. angezeigt. Im unteren Textfeld wird der Übersetzungsfortschritt der gerade in Bearbeitung befindlichen Zeile des Roboterprogramms angezeigt. Nach der Übersetzung wird noch die Zeit angezeigt, die für die Übersetzung des Roboterprogramms benötigt wurde. In Abb. Bild B9 ist ein Ausschnitt des Fensters der Off-line-Programmierung dargestellt, während die Übersetzung durchgeführt wird. Im ersten Textfeld wird angezeigt, dass die Übersetzung von Zeile 3 erfolgreich war (*line 3: ok*). Dem zweiten Textfeld kann entnommen werden, dass 22 % der aktuellen Zeile (Zeile 4) des Roboterprogramms übersetzt wurden. Die Zeile, die gerade übersetzt wird, ist im Roboterprogramm blau hervorgehoben.

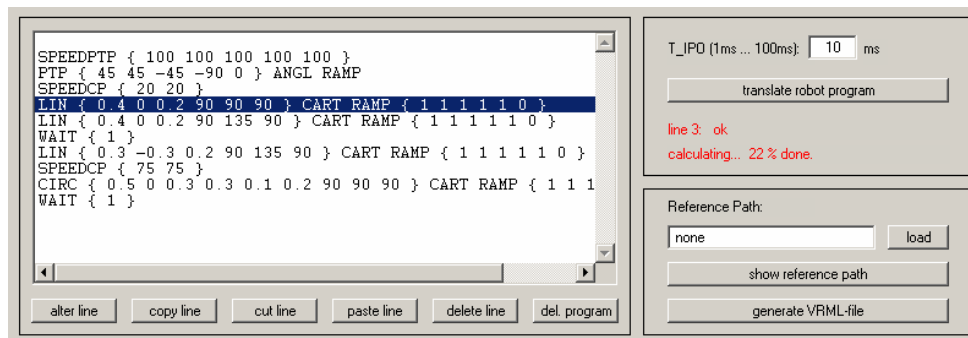


Bild B9 Aufruf zur Übersetzung des Roboterprogramms

Nach der erfolgreichen Übersetzung des Roboterprogramms erscheint ein Dialog, in dem ein Pfad und ein Dateiname für die Sollwerte angegeben werden muss. Der standardmäßig vorgegebene Pfad `...\mandy\referencepath` sollte beibehalten werden, da bei der späteren Auswahl der Sollwerte in der Off-line-Programmierung oder der Simulationsumgebung wieder standardmäßig dieser Pfad vorgeschlagen wird.

Aufgrund des in ManDy verwendeten Algorithmus für die Rückwärts-transformation, kann es bei bestimmten Roboterstellungen teilweise zu unerwarteten Ergebnissen kommen. Bei der Erstellung des Roboterprogramms sollte deshalb darauf geachtet werden, dass eine CP-Bahn niemals am Rand des Arbeitsbereichs des Roboters begonnen wird. Der TCP des Roboters sollte erst mit einem beliebigen PTP-Befehl vom Rand des Arbeitsbereiches wegbewegt werden.

Bei PTP-Befehlen mit kartesischen Zielstellungen sollte darauf geachtet werden, dass sich die kartesische Zielstellung nicht all zu weit von der Startstellung entfernt befindet. Wird dies nicht beachtet kann es vorkommen, dass obwohl die Zielstellung vom Roboter erreicht werden kann, die Fehlermeldung *out of workspace* angezeigt wird.

Visualisierung

In der Off-line-Programmierungsumgebung muss zunächst die Parameterdatei des Roboters geladen werden, für den die Sollwerte angezeigt werden sollen. Danach wird die Datei der anzuzeigenden Sollwerte über den *load*-Button (im Fenster rechts unten, siehe Bild B9) geladen. Bei Betätigung des Buttons *show reference path* öffnet sich ein neues Fenster. In diesem Fenster können die Sollwerte in verschiedenen Ansichten dargestellt werden. Die Auswahl der gewünschten Ansicht erfolgt über vier Popup-Menüs. Die Inhalte der Popup-Menüs sind in Bild B10 beispielhaft für einen Roboter mit vier Gelenken dargestellt.

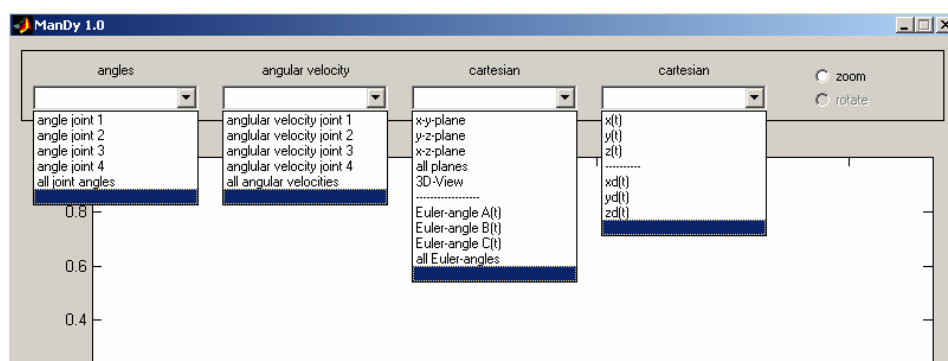


Bild B10 Grafische Darstellung des Sollverlaufes

Im ersten Popup-Menü kann man sich die zeitlichen Verläufe Sollwerte der Gelenkkoordinaten anzeigen lassen. Es können entweder einzelne Gelenkkoordinaten angezeigt werden oder alle Gelenkkoordinaten zusammen. Über das zweite Popup-Menü kann die Darstellung der zeitlichen Verläufe Gelenkgeschwindigkeiten erfolgen. Auch hier können die Geschwindigkeiten einzelner Gelenke oder die Geschwindigkeiten aller Gelenke ausgewählt werden.

Im dritten Popup-Menü sind verschiedene kartesische Ansichten zusammengefasst. Die Sollwerte der Raumbahn können in verschiedenen Ebenen oder als 3D-Darstellung angezeigt werden. Weiterhin können die zeitlichen Verläufe der Sollwerte der EULER-Winkel entweder einzeln oder zusammen dargestellt werden.

Das vierte Popup-Menü beinhaltet weitere kartesische Darstellungen. Hier kann man sich die einzelnen kartesischen Koordinaten über der Zeit anzeigen lassen. Außerdem können die Geschwindigkeiten in den drei Raumrichtungen über der Zeit dargestellt werden.

Über die beiden Radio-Buttons *zoom* und *rotate* im Fenster rechts oben (siehe Bild B10) kann die gerade angezeigte Darstellung vergrößert und gedreht werden. Zur Vergrößerung einzelner Ausschnitte betätigt man zuerst den Radio-Button *zoom* und zieht dann, bei gedrückt gehaltener linker Maustaste, ein Rechteck um den gewünschten Plot-Bereich. Der ausgewählte Bereich wird dann vergrößert dargestellt. Das Drehen (*rotate*) ist nur bei der 3D-Darstellung der Raumbahn möglich. Man wählt zunächst den Radio-Button *rotate* aus, klickt auf die 3D-Darstellung und zieht das Bild, bei gedrückt gehaltener linker Maustaste, in die gewünschte Orientierung.

Eine weitere Möglichkeit der Darstellung der Sollwerte ist die Animation durch das VRML-Modell des Roboters. Mit einem Klick auf den Button *generate VRML-file* wird automatisch ein einfaches 3D-Modell des Roboters erzeugt, das die programmierten Sollbewegungen durchführt (siehe Bild B11). Die Erzeugung des VRML-Modells kann je nach

Größe des Roboterprogramms und nach der Leistungsfähigkeit des Computers bis zu einer Minute in Anspruch nehmen. Im Normalfall sollte das Modell aber nach wenigen Sekunden zu sehen sein. Voraussetzung für die Darstellung des VRML-Modells ist wie in Abschnitt B2 erwähnt ein installierter VRML-fähiger Web-Browser und die Angabe des korrekten Pfades unter *options*.

Das VRML-Modell des Roboters kann aus verschiedenen Perspektiven betrachtet werden. Klickt man mit der rechten Maustaste auf die Darstellung des Roboters, so erscheint ein Kontextmenü. Über den obersten Menüpunkt *Viewpoints* können drei vordefinierte Einstellungen angewählt werden. Es sind dies die Ansichten der drei Ebenen $x-z$, $y-z$ und $x-y$. Weiterhin kann über die vom VRML-Plug-In zur Verfügung gestellten Buttons navigiert werden. Beim Cortona-Plug-In können die in Bild B11 dargestellten Buttons *walk*, *fly*, *study*, *plan*, *pan*, *turn*, *roll*, *goto*, *align*, *view*, *restore* und *fit* verwendet werden. Beim Cosmo-Plug-In haben die Buttons z. T. andere Namen für dieselben oder ähnliche Funktionen.

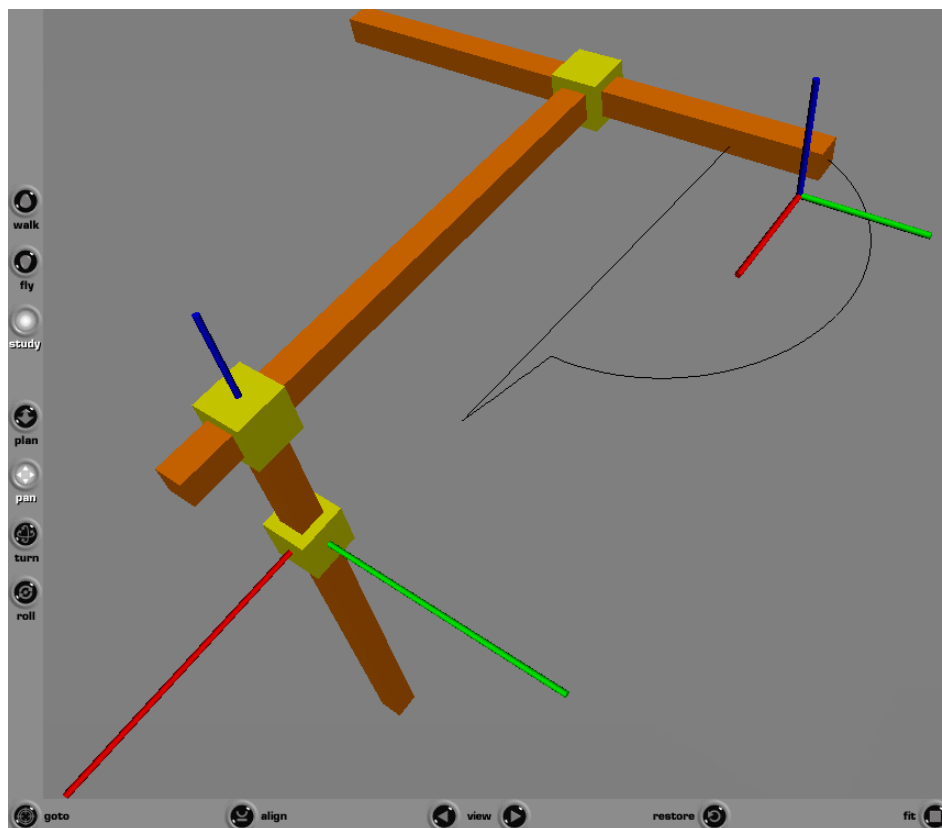


Bild B11 VRML-Modell eines TTT-Roboters

B4 Simulation des Regelungsverhaltens

Wählt man in ManDy den Menüpunkt *simulation* an, erscheint das in Bild B12 dargestellte Fenster. Bevor die Simulation gestartet werden kann, müssen jedoch verschiedene Einstellungen vorgenommen werden. Zunächst muss die Parameterdatei des Roboters geladen werden, für den die Simulation durchgeführt werden soll. Über den *load*-Button kann ein Dialogfenster geöffnet werden, in dem die Parameterdatei ausgewählt werden kann. Es wird automatisch in den Standardpfad für Roboterparameterdateien `...\mandy\robots` gewechselt. Danach müssen Parameterdateien für das Robotermodell (*robot model*), für die Regelungsparameter (*control parameters*) und für die Simulationsparameter (*simulation parameters*) geladen werden. Über die *load*-Buttons können auch hier Dialogfenster geöffnet werden, in denen die Parameterdateien ausgewählt werden können. Es wird automatisch in die Standardpfade der Parameterdateien `...\mandy\robotmodel`, `...\mandy\controlpara` bzw. `...\mandy\simulationpara` gewechselt. Sollten die nötigen Parameterdateien noch nicht vorliegen, so müssen sie erst erstellt werden. Die Erstellung von Parameterdateien wird in den nächsten Abschnitten erläutert.

Als letztes muss die Sollbahn (*reference path*) geladen werden. Über den *load*-Button kann das Dialogfenster geöffnet werden, in dem die Datei der Sollwerte geladen werden kann, die schon programmiert worden ist (s. Abschn. B3). Es wird automatisch in den Standardpfad für Sollwertdateien `...\mandy\referencepath` gewechselt.

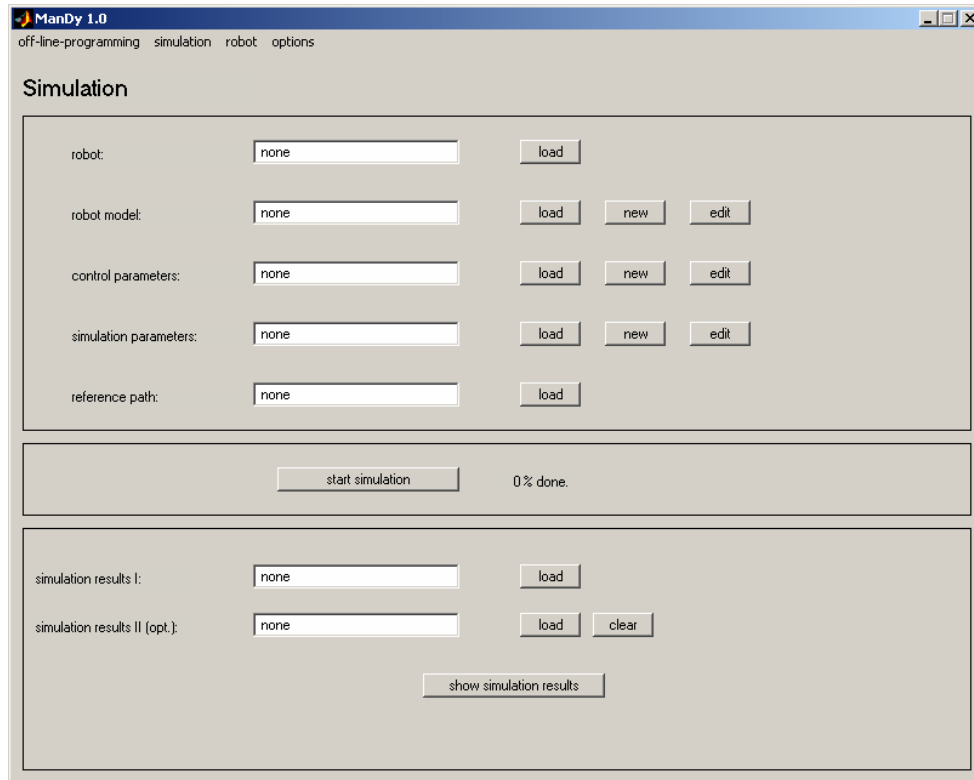


Bild B12 Startfenster für die Simulation des Regelungsverhaltens

Nachdem alle Parameterdateien geladen wurden, kann die Simulation über den Button *start simulation* gestartet werden. Der Simulationsfortschritt wird neben dem *start simulation*-Button angezeigt. Nach der Beendigung der Simulation können die Simulationsergebnisse in einer weiteren Datei gespeichert werden. Nun werden die verschiedenen benötigten Parametersätze erläutert.

Modellparameter

Modellparameter werden nur benötigt, wenn ein modellbasierter Regler verwendet werden soll. Bei Einzelgelenkregelungen kann das Feld frei bleiben. Über den Button *new* bzw. *edit* kann das Fenster, in dem Parameter eingetragen werden können, geöffnet werden. Mit dem *new*-Button öffnet sich ein leeres Fenster (Bild B13). Klickt man auf den *edit*-Button, so muss in einem Dialog eine bereits bestehende Parameterdatei ausgewählt werden, die danach abgeändert werden kann.

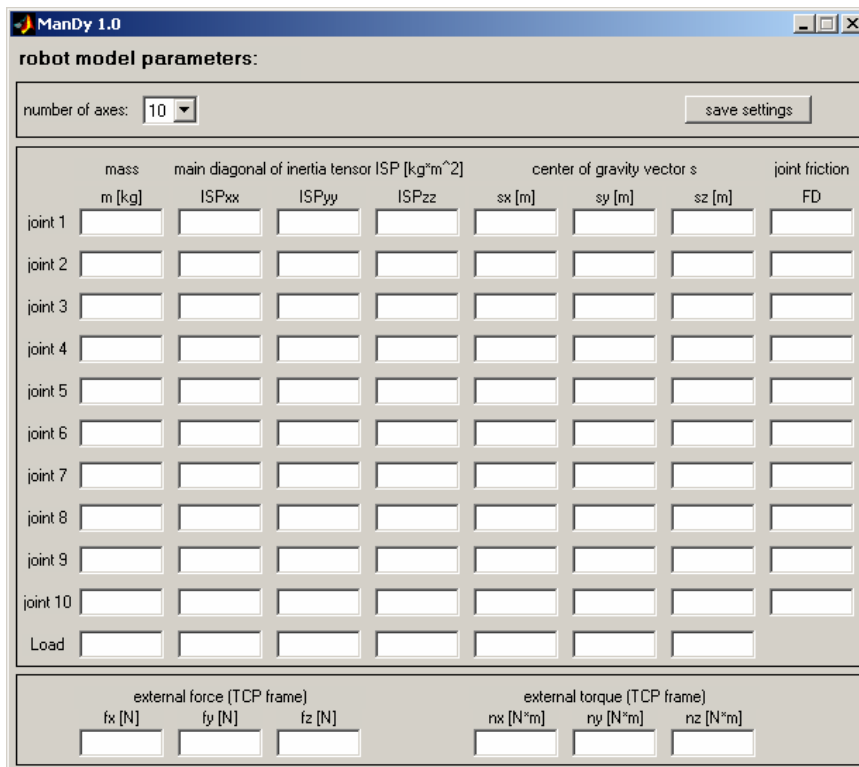


Bild B13 Parameter der Dynamik

Hier besteht die Möglichkeit Modellungenauigkeiten zu simulieren. D.h. ein modellbasierter Regler nach Kap. 7.4 verwendet nicht exakt die Parameter der Dynamik wie sie für das Modell des Roboters nach B2 (s. Bilder B4, B5) eingegeben wurden, sondern man kann Abweichungen annehmen, um die Robustheit der modellbasierten Regelungsverfahren zu testen. In Tabelle B4 und Bild B13 sind diese Parameter aufgeführt.

Tabelle B4: Parameter der Dynamik, die von der Regelung verwendet werden

Parameter	Beschreibung	Einheit
m	Armteilmasse	kg
$ISP_{xx}, ISP_{yy}, ISP_{zz}$	Hauptdiagonale des Trägheitstensors	kg·m ²
s_x, s_y, s_z	Schwerpunktsvektor	m
FD	abtriebsseitiger Reibungskoeffizient	N·m·s ^o oder N·s/m
f_x, f_y, f_z	Externe Kraft	N
n_x, n_y, n_z	Externes Moment	N·m

Die Anzahl der Gelenke ist zuerst einzustellen. Danach können die einzelnen Parameter eingetragen werden. Mit dem Button *save settings* können die Parameter in einer Datei

gespeichert werden. In der Zeile load (Last) können die Parameter einer Lastmasse eingetragen werden. Die Werte beziehen sich auf das Werkzeugkoordinatensystem K_N .

Regelungsparameter

Auch für die in der Simulation zu verwendenden Regelungsparameter muss eine Parameterdatei erstellt werden. Mit dem *new*-Button öffnet sich ein leeres Fenster (Bild B14). Hier können die Regelungsparameter eingetragen werden. Zunächst sollte wieder die Anzahl der Robotergelenke eingestellt werden, danach der zu verwendende Regler. Es sind die in Kap. 7 diskutierten Regelverfahren auswählbar. Im Parameterfeld sind dann die geforderten Regelungsparameter vollständig einzutragen und über den Button *save settings* in einer Datei zu speichern.

	KL	Kv	KP	TN
joint 1	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 2	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 3	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 4	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 5	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 6	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 7	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 8	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 9	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
joint 10	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

Bild B14 Regelungsparameter

Mit Hilfe des *edit*-Buttons in der Simulationsumgebung kann auch hier eine bereits existierende Parameterdatei geladen und verändert werden.

Simulationsparameter

Mit dem *new*- bzw. *edit*-Button kann das Dialogfenster zum Eintragen neuer bzw. zum Abändern bereits existierender Simulationsparameter geöffnet werden (siehe Bild B15).

Die Simulationsparameter sind im Gegensatz zu den Modell- und Regelungsparametern nicht roboterspezifisch. D. h. sie können für jeden beliebigen Roboter verwendet werden.

Über die Radio-Buttons kann das gewünschte Lösungsverfahren (*solver*) für die Bewegungsgleichung des Roboters ausgewählt werden. Sollen die Getriebe des Roboters während der Simulation als ideal steif betrachtet werden, so kann zwischen einem RUNGE-KUTTA-Verfahren 2. Ordnung (*RK2 rigid*) und einem RUNGE-KUTTA-Verfahren 4.

Ordnung (*RK4 rigid*) gewählt werden. Möchte man die Getriebe elastisch simulieren, so steht ein **RUNGE-KUTTA-Verfahren 4. Ordnung (*RK4 elastic*)** zur Verfügung.

Der Regelungstakt kann über die Abtastzeit (*sampling time*) festgelegt werden. Dabei ist zu beachten, dass die Abtastzeit T_A in jedem Fall kleiner oder gleich der Interpolationszeit T_{ipo} , mit der die Bahnplanung durchgeführt wurde, sein muss. Der Standardwert für die Abtastzeit beträgt 0.001 Sekunden.

Es kann eine Lastmasse eingetragen werden. Die Lastmasse wird in einem eigenen Koordinatensystem beschrieben, das dem Werkzeugkoordinatensystem entspricht. Die Angaben zum Trägheitstensor und Schwerpunktsvektor beziehen sich auf dieses Koordinatensystem.

Evtl. auftretende Bearbeitungskräfte und Bearbeitungsmomente können ebenfalls eingetragen werden. Diese Kräfte und Momente müssen bezüglich des Werkzeugkoordinatensystems angegeben werden.

ManDy 1.0

simulation parameters:

Solver: RK2 rigid
 RK4 rigid
 RK4 elastic

mass Load [kg] sampling time [s]

main diagonal inertia tensor Load
ISPxx kg*m² ISPyy kg*m² ISPzz kg*m²

center of gravity vector Load (tool coordinate system)
sx [m] sy [m] sz [m]

external force (tool coordinate system)
fx [N] fy [N] fz [N]

external torque (tool coordinate system)
nx [N*m] ny [N*m] nz [N*m]

save settings

Bild B15 Simulationsparameter

Mit dem Button *save settings* können die Simulationsparameter in einer Datei gespeichert werden.

Anzeigen der Simulationsergebnisse

Im unteren Teil der Simulationsumgebung (siehe Bild B12) befinden sich die Einstellmöglichkeiten für die Darstellung der Simulationsergebnisse. Mit Hilfe des oberen *load*-Buttons kann ein zuvor abgespeichertes Simulationsergebnis ausgewählt werden (*simulation results I*). Es ist auch möglich, zwei Simulationsergebnisse miteinander zu vergleichen. Dazu muss über den unteren *load*-Button eine weitere Datei mit Simulationsergebnissen ausgewählt werden (*simulation results II*). Ein Vergleich von Simulationsergebnissen macht natürlich nur Sinn, wenn beide Simulationen mit den gleichen Sollwerten durchgeführt wurden. Bei unterschiedlichen Sollwerten wird deshalb eine Fehlermeldung angezeigt.

Bei Betätigung des Buttons *show simulation results* öffnet sich ein neues Fenster. In diesem Fenster können die Simulationsergebnisse in verschiedenen Ansichten dargestellt werden. Die Auswahl der gewünschten Ansicht erfolgt über fünf Popup-Menüs. Die Inhalte der Popup-Menüs sind in Bild B16 beispielhaft für einen Roboter mit vier Gelenken dargestellt.

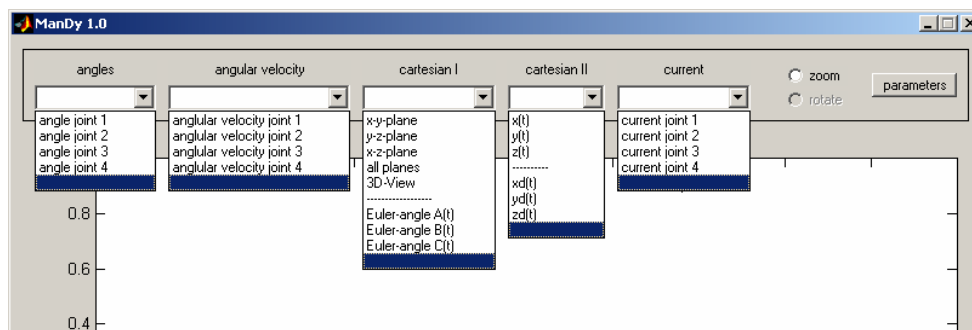


Bild B16 Darstellung der Simulationsergebnisse

Im ersten Popup-Menü können die einzelnen Gelenkkoordinaten ausgewählt werden. Dargestellt werden jeweils der Sollwert sowie die simulierten Istwerte der Antriebs- und Abtriebsseite über der Zeit. Über das zweite Popup-Menü kann die Darstellung der Gelenkgeschwindigkeiten erfolgen. Angezeigt werden die Sollgeschwindigkeit aus der Bahnplanung, die Sollgeschwindigkeit aus der Ausgangsgröße des Lagereglers sowie die simulierten Istwerte der antriebsseitigen und abtriebsseitigen Geschwindigkeiten. Im dritten und vierten Popup-Menü sind verschiedene kartesische Ansichten zusammengefasst. Auch hier werden jeweils der Sollwert und die simulierten Istwerte der Antriebs- und Abtriebsseite angezeigt. Im letzten Popup-Menü kann die Darstellung der einzelnen Stellwerte (Sollwerte der Motorströme ausgewählt werden). Über die beiden Radio-Buttons *zoom* und *rotate* im Fenster rechts oben (siehe Bild B16) kann die gerade angezeigte Darstellung vergrößert und rotiert werden. Die Vorgehensweise wurde bereits in Abschnitt B3, Bild B10 erläutert. Mit dem *parameters*-Button (in Bild B16 rechts oben) können die, bei der Simulation, verwendeten Parameter angezeigt werden. Die Roboterparameter, Modellparameter, Regelungsparameter und Simulationsparameter werden in tabellarischer Form dargestellt. Diese Funktion ist sehr nützlich, wenn bestimmte Simulationsergebnisse rekonstruiert werden sollen.