(19) Deutsches Patent- und Markenamt

(10) **DE 11 2021 003 199 B4** 2025.07.10

(12) # Patentschrift

(21) Deutsches Aktenzeichen: **11 2021 003 199.6**
(86) PCT-Aktenzeichen: **PCT/EP2021/065699**
(87) PCT-Veröffentlichungs-Nr.: **WO 2021/250201**
(86) PCT-Anmeldetag: **10.06.2021**
(87) PCT-Veröffentlichungstag: **16.12.2021**
(45) Veröffentlichungstag der Patenterteilung: **10.07.2025**

(51) Int Cl.: ***B25J 9/16*** (2006.01)
B25J 13/00 (2006.01)

Innerhalb von neun Monaten nach Veröffentlichung der Patenterteilung kann nach § 59 Patentgesetz gegen das Patent Einspruch erhoben werden. Der Einspruch ist schriftlich zu erklären und zu begründen. Innerhalb der Einspruchsfrist ist eine Einspruchsgebühr in Höhe von 200 Euro zu entrichten (§ 6 Patentkostengesetz in Verbindung mit der Anlage zu § 2 Abs. 1 Patentkostengesetz).

(66) Innere Priorität
10 2020 115 517.4 10.06.2020

(73) Patentinhaber:
Deutsches Zentrum für Luft- und Raumfahrt e.V., 53227 Bonn, DE

(74) Vertreter:
EGE LEE & ROIDER Patentanwälte PartGmbB, 80807 München, DE

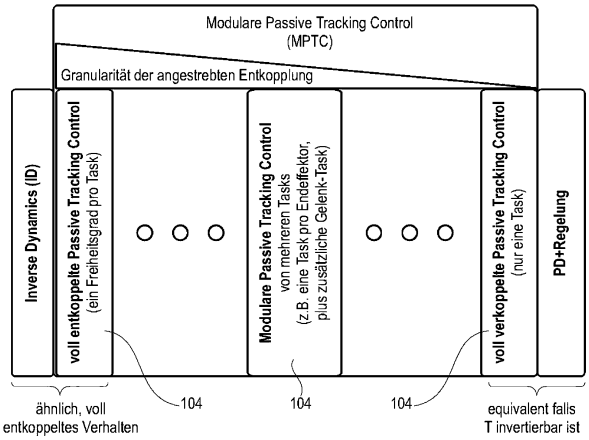
(72) Erfinder:
Engelsberger, Johannes, 82234 Weßling, DE; Garofalo, Gianluca, 82234 Weßling, DE; Mesesan, George Adrian, 82234 Weßling, DE

(56) Ermittelter Stand der Technik:

DE	**10 2010 064 267**	**B4**
DE	**10 2015 106 227**	**B3**
FR	**3 054 048**	**B1**
EP	**1 972 416**	**B1**
EP	**2 740 012**	**B1**
CN	**1 10 799 308**	**A**

(54) Bezeichnung: **Verfahren und Computerprogrammprodukt zum Regeln eines Roboters**

(57) Hauptanspruch: Verfahren zum Regeln eines kinematisch redundanten Roboters (100), um mehrere Tasks zu erfüllen, wobei wenigstens ein passivitäts-basiertes erstes Reglermodul (102) verwendet wird, für das wenigstens eine erste Reglermodul (102) wenigstens eine Task-Zielbeschreibung und wenigstens ein zugehöriges Task-Mapping berechnet werden, für die Tasks wenigstens eine Gewichtung in Form einer task-spezifischen symmetrischen Matrix $W = \Psi_k M_k^{-1}$ berechnet wird, wobei sich die Matrix W_k aus der Inversen der Task-Trägheitsmatrix M_k und einem Gewichtungs-Skalar Ψ_k zur unabhängigen Gewichtung verschiedener Tasks untereinander zusammensetzt, und das wenigstens eine erste Reglermodul (102) unter Verwendung der wenigstens einen Task-Gewichtungs-Matrix W_k, welche in Form von Sub-Matrizen in eine symmetrische und block-diagonale Gesamt-Gewichtungs-Matrix W zusammengeführt wird, in einen Gesamtregler (104) integriert wird.



Beschreibung

[0001] Die Erfindung betrifft ein Verfahren zum Regeln eines kinematisch redundanten Roboters, um mehrere Tasks zu erfüllen. Außerdem betrifft die Erfindung ein Computerprogrammprodukt zum Ausführen eines derartigen Verfahrens.

[0002] Die simultane Regelung mehrerer Aufgaben, sogenannter „Tasks“, stellt ein wichtiges Forschungsthema auf dem Gebiet der Roboter-Regelung dar. Während frühere Arbeiten den einfacheren Fall der Regelung eines einzigen Tasks und des entsprechenden Nullraumes, also der verbleibenden Roboter-Freiheitsgrade, für kinematisch redundante Roboter betrachteten, so existieren heute mehrere etablierte Methoden, welche die Behandlung mehrerer Tasks mit und ohne Prioritäten ermöglichen. Die Literatur unterscheidet Arbeiten, die das Task-Koordinationsproblem zunächst auf kinematischer Ebene lösen, von Arbeiten, welche das Regelungsproblem direkt auf der Dynamik-Ebene behandeln. Eine weitere wichtige Klassifikationsmöglichkeit besteht in der Einteilung in Verfahren, welche durch hierarchische Regler strikte Task-Prioritäten erzwingen, und andere Verfahren, welche eine softe oder weiche Priorisierung durch Task-Gewichtung erzeugen.

[0003] Auf der Kinematik-Ebene wurden mit den Veröffentlichungen Y. Nakamura, H. Hanafusa, and T. Yoshi Kawa. Taskpriority based redundancy control of robot manipulators. The International Journal of Robotics Research, 6(2): 3-15, 1987. doi: 10.1177/027836498700600201. und G. Antonelli. Stability Analysis for Prioritized closedloop inverse kinematic algorithms for redundant robotic systems. IEEE Transactions on Robotics, 25(5):985-994, 2009. hierarchische Regler vorgeschlagen, welche auf Nullraum-Projektionen beruhen und eine strikte Task-Hierarchie garantieren. Um dem Problem möglicher Task-Singularitäten zu begegnen, wurde in der Veröffentlichung F. Kanehiro, H. Hirukawa, and S. Kajita. Openhrp: Open Architecture Humanoid Robotics Platform. The International Journal of Robotics Research, 23(2):155- 165, 2004. doi: 10.1177/0278364904041324. eine singularitätsrobuste Inversekinematikmethode vorgeschlagen. Allerdings geht durch diese singularitäts-robuste Inverskinematik die strikte Task-Hierarchie verloren, da hierdurch im Endeffekt eine Gewichtung zwischen den einzelnen Tasks erzeugt wird.

[0004] Andere Verfahren behandeln mehrere Tasks gleichzeitig auf der Dynamik-Ebene. Das „Operational Space“ Verfahren wurde weiterentwickelt, um die Regelung humanoider Roboter zu ermöglichen. Dazu wird auf die Veröffentlichungen L. Sentis and O. Khatib. Synthesis of Whole-Body behaviors through hierarchical control of behavioral primitives. International Journal of Humanoid Robotics, 2(4): 505-518, 2005. und L. Sentis, J. Park, and O. Khatib. Compliant Control of multicontact and center-of-mass behaviors in humanoid robots. IEEE Transactions on Robotics, 26(3):483-501, 2010. verwiesen. Andere auf „Inverse Dynamics“ (ID) basierende Verfahren verwenden hierarchische Quadratische Programme (QP). Dazu wird auf die Veröffentlichungen J. Peters, M. Mistry, F. Udwadia, J. Nakanishi, and S. Schaal. A unifying framework for robot control with redundant dofs. Autonomous Robots, 24:1-12, 2008. doi: 10.1007/s10514-007-9051-x., Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: fast online humanoid-robot motion generation. The International Journal of Robotics Research, 33(7):1006-1028, 2014. doi: 10.1177/0278364914521306. und K. Bouyarmane and A. Kheddar. On Weight-Prioritized multitask control of humanoid robots. IEEE Transactions on Automatic Control, 63(6):1632-1647, June 2018. ISSN 2334-3303. doi: 10.1109/TAC.2017.2752085. verwiesen. Die meisten der genannten Verfahren zielen auf eine strikte Task-Entkopplung ab. Hierdurch wird beispielsweise, zumindest in der Theorie, sichergestellt, dass sich unterschiedliche Tasks nicht gegenseitig beeinflussen, und ebenso das jeweilige Einschwingverhalten einzelner Tasks nach Störungen völlig unabhängig von anderen Task-Koordinaten ist.

[0005] Das hier vorgestellte Verfahren ist durch die Familie von Inverse Dynamics basierten Tracking-Reglern inspiriert, welche durch die Verwendung eines einzigen gewichteten Quadratischen Programmes (QP) einen sanften Kompromiss zwischen einer Reihe von Tasks anstreben. Dazu wird auf die Veröffentlichungen M. A. Hopkins, D. W. Hong, and A. Leonessa. Compliant locomotion using whole-body control and divergent component of motion tracking. In IEEE Int. Conf. on Robotics and Automation (ICRA), pages 5726-5733, May 2015. doi: 10.1109/ICRA.2015.7140001., T. Koolen, S. Bertrand, G. Thomas, T. de Boer, T. Wu, J. Smith, J. Englsberger, and J. Pratt. Design of a Momentum-Based Control Framework and application to the humanoid robot atlas. International Journal of Humanoid Robotics, 13(01):1650007, 2016. doi: 10.1142/S0219843616500079. Und J. Englsberger, G. Mesesan, A. Werner, and C. Ott. Torque-Based Dynamic Walking - a long way from simulation to experiment. In IEEE Int. Conf. on Robotics and Automation (ICRA), pages 440-447, 2018. verwiesen. Solche Regler sind sehr einfach zu implementieren und zeichnen sich durch ihre hohe Flexibilität aus. Im Vergleich zu passivitäts-basierten Verfahren gemäß den Veröffentlichungen B. Paden and B. Riedle. A Positive-Real Modification of a class of nonlinear controllers for robot manipulators. In 1988 American Control Conference, pages 1782-1785, June 1988. doi: 10.23919/ACC.1988.4790015., A. Dietrich. Whole-Body Impedance Control of Wheeled

Humanoid Robots, volume 116. Springer International Publishing, 2016. ISBN 978-3-319-40557-5., B. Henze, M. A. Roa, and Ch. Ott. Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios. The International Journal of Robotics Research, 35 (12):1522-1543, 2016. doi: 10.1177/0278364916653815., G. Mesesan, J. Englsberger, G. Garofalo, C. Ott, and15:47 10.06.2020 A. Albu-Schäffer. Dynamic walking on compliant and uneven terrain using dcm and passivity-based wholebody control. In IEEE-RAS 19th Int. Conf. on Humanoid Robots (Humanoids), pages 25-32, 2019. und A. Dietrich and C. Ott. Hierarchical Impedance-Based tracking control of kinematically redundant robots. IEEE Transactions on Robotics, 36 (1):204-221, 2020. ISSN 1941-0468. doi: 10.1109/TRO.2019.2945876. sind Inverse Dynamics basierte Regler allerdings weniger robust in Bezug auf Modellfehler und Kontaktunsicherheiten, sodass im realen Einsatz Probleme wie beispielsweise Vibrationen auftreten, welche dann wiederum mittels heuristischer Methoden behandelt werden müssen.

[0006] Weiterhin ist das vorgestellte Verfahren sowohl durch den gewichtungs-basierten Multitasking Regler aus der Veröffentlichung K. Bouyarmane and A. Kheddar. On Weight-Prioritized multitask control of humanoid robots. IEEE Transactions on Automatic Control, 63(6):1632-1647, June 2018. ISSN 2334-3303. doi: 10.1109/TAC.2017.2752085. inspiriert, als auch durch den strikt hierarchischen passivitäts-basierten Regler aus der Veröffentlichung A. Dietrich and C. Ott. Hierarchical Impedance-Based tracking control of kinematically redundant robots. IEEE Transactions on Robotics, 36(1):204-221, 2020. ISSN 1941-0468. doi: 10.1109/TRO.2019.2945876. Ähnlich wie in der Veröffentlichung K. Bouyarmane and A. Kheddar. On Weight-Prioritized multitask control of humanoid robots. IEEE Transactions on Automatic Control, 63 (6):1632-1647, June 2018. ISSN 2334-3303. doi: 10.1109/TAC.2017.2752085. verwendet das vorgestellte Verfahren ein Quadratisches Programm (QP), um die Regelungsziele der einzelnen Tasks zu vereinigen bzw. gegeneinander abzuwägen. Hierbei wird allerdings in der Veröffentlichung K. Bouyarmane and A. Kheddar. On Weight-Prioritized multitask control of humanoid robots. IEEE Transactions on Automatic Control, 63(6):1632-1647, June 2018. ISSN 2334-3303. doi: 10.1109/TAC.2017.2752085.jeder der zu den einzelnen Tasks gehörigen Regler basierend auf Inverse Dynamics berechnet, wobei eine Einheitsmatrix als gewünschte Trägheit verwendet wird, was einer Feedback Linearisierung entspricht. Im Gegensatz hierzu berechnet das vorgestellte Verfahren die zu den einzelnen Tasks gehörigen Regler basierend auf dem Konzept der Passivität und verwendet die natürliche Trägheit des Roboters, so dass ein Verhalten ähnlich dem eines PD+ Reglers erzielt wird. Im Gegensatz zum aus der Veröffentlichung A. Dietrich and C. Ott. Hierarchical Impedance-Based tracking control of kinematically redundant robots. IEEE Transactions on Robotics, 36(1):204-221, 2020. ISSN 1941-0468. doi: 10.1109/TRO.2019.2945876. bekannten Verfahren, welches ebenfalls auf der Verwendung der natürlichen Trägheit für den Regelungsansatz basiert, verwendet das vorgestellte Verfahren eine gewichtete QP Formulierung, also eine weiche Priorisierung, wodurch es möglich wird, eine beliebige Anzahl von verschiedenen Tasks zu kombinieren, wobei das Verfahren außerdem in bestimmten Situationen, wie beispielsweise beim Auftreten einer Task-Singularität, zu einem weniger aggressiven Verhalten führt.

[0007] Die DE 10 2015 106 227 B3 beschreibt ein Verfahren und eine Vorrichtung zum Steuern und Regeln von Motoren eines Roboters. Die DE 10 2010 064 267 B4 beschreibt eine Steuer- bzw. Regeleinrichtung für einen mobilen Roboter. Die FR 3 054 048 B1 beschreibt ein Verfahren zur Echtzeitsteuerung einer gelenkigen Massenstruktur. Die EP 2 740 012 B1 beschreibt einen Roboter mit Gelenken mit variabler Steifigkeit und ein Verfahren zur Berechnung einer optimierten Steifigkeit. Die EP 1 972 416 B1 beschreibt ein Verfahren zur Steuerung eines Roboters. Die CN 1 10 799 308 A beschreibt Techniken zur effektiven Bestimmung von Steuerungsparametern für einen physischen oder simulierten Roboter.

[0008] Der Erfindung liegt die Aufgabe zugrunde, ein eingangs genanntes Verfahren zu verbessern. Außerdem liegt der Erfindung die Aufgabe zugrunde, ein eingangs genanntes Computerprogrammprodukt bereitzustellen.

[0009] Die Aufgabe wird gelöst mit einem Verfahren mit den Merkmalen des Anspruchs 1. ußerdem wird die Aufgabe gelöst mit einem Computerprogrammprodukt mit den Merkmalen des Anspruchs Fehler! Verweisquelle konnte nicht gefunden werden. Vorteilhafte Ausführungen und/oder Weiterbildungen sind Gegenstand der Unteransprüche.

[0010] Das Verfahren kann zum regelungstechnischen Kontrollieren des Roboters dienen. Das Verfahren kann beispielsweise in der Servicerobotik, in der Medizinrobotik, in der Industrie 4.0, im Bereich einer robotischen Assistenz für Astronauten eingesetzt werden. Das Verfahren kann grundsätzlich überall dort eingesetzt werden, wo kraft-/drehmomentgeregelte Roboter mit kinematischer Redundanz eingesetzt werden können und sowohl eine hohe Präzision als auch eine nachgiebige Mensch-Roboter-Interaktion bzw. Umgebungs-Roboter-Interaktion gewünscht sind.

[0011] Ein kinematisch redundanter Roboter kann derart viele Freiheitsgrade aufweisen, dass mehrere Aufgaben oder Tasks gleichzeitig umgesetzt werden können. Die Tasks können voneinander unabhängig sein. Die Tasks können zumindest teilweise gleichzeitig ausführbar sein. Der Roboter kann kraftgeregelt und/oder drehmomentgeregelt sein. Das passivitäts-basierte Reglermodul kann ein impedanzbasiertes Reglermodul sein. Der Begriff „Passivität“ wird vorliegend insbesondere zur Abgrenzung einer „Aktivität“ verwendet, bei der eine ungewollte Energiezunahme im System auftritt.

[0012] Der Begriff „Task-Zielbeschreibung“ beschreibt vorliegend insbesondere eine Größe, welche sich aus einer Kombination von Termen ergibt, die der eigentlichen Zielsetzung des jeweiligen Tasks entsprechen, mit entsprechenden Vorsteuerungs- bzw. Kompensations-Termen. Die entsprechende Zielsetzung des Tasks kann beispielsweise aus einem nominellen Tracking einer Referenztrajektorie und/oder auch, vorteilhafterweise gleichzeitig, aus dem nachgiebigen Impedanzverhalten, beispielsweise im Falle einer Interaktion mit der Umgebung, einem Objekt oder einem Menschen, bestehen. Die Vorsteuerungs- und/oder Kompensations-Terme können beispielsweise Gravitations-Effekte sowie Coriolis- und/oder Zentrifugal-Effekte ganz oder teilweise kompensieren, um das eigentliche Wunsch-Verhalten und/oder die eigentliche Wunsch-Dynamik zu erzeugen.

[0013] Der Begriff „Task-Mapping“ beschreibt vorliegend insbesondere eine Größe, welche den Zusammenhang zwischen den gegebenen Stellgrößen und/oder Regelungsinputs und der jeweiligen vorgenannten Task-Zielbeschreibung herstellt. Die Stellgrößen und/oder Regelungsinputs können beispielsweise aus stellbaren Drehmomenten in den Roboter-Gelenken, sowie, im Falle von free-floating Robotern wie zum Beispiel humanoiden Robotern, auch aus Kontaktkräften bestehen, welche sich unter bestimmten Voraussetzungen aus den Zwangsbedingungen zwischen den Gelenk-Drehmomenten und gegebenen Endeffekor-Beschleunigungen, also beispielsweise Roboter-Fuß-Beschleunigungen, ergeben.

[0014] Die wenigstens eine Task-Zielbeschreibung und das wenigstens eine zugehörige Task-Mapping können derart berechnet werden, dass ein nominelles Verhalten des wenigstens einen ersten Reglermoduls dem Verhalten eines Feder-Masse-Dämpfer-Systems entspricht.

[0015] Die wenigstens eine Task-Zielbeschreibung und das wenigstens eine zugehörige Task können basierend auf einer natürlichen Roboter-Trägheit berechnet werden.

[0016] Die wenigstens eine Task-Zielbeschreibung und das wenigstens eine zugehörige Task-Mapping können derart berechnet werden, dass taskspezifische Coriolis- und Zentrifugal-Effekte ungeändert bleiben, während alle übrigen Coriolis- und Zentrifugal-Effekte kompensiert werden.

[0017] Die wenigstens eine Task-Zielbeschreibung kann als Task-Vektor berechnet werden. Der Task-Vektor kann eine Kraft, eine Kraftrichtung, ein Moment und/oder eine Momentrichtung angeben. Die gewünschte Task-Kraft $f_{k,des}$in Gleichung (17) stellt einen solchen Task-Vektor dar. Weiterhin können die einzelnen Task-Vektoren vorteilhafterweise zu einem kombinierten Task-Vektor zusammengefasst werden. Die in Gleichung (21) beschriebene „Kombination aller gewünschten Task-Kräfte“ f_{des} stellt ein Beispiel für einen solchen kombinierten Task-Vektor dar.

[0018] Das wenigstens eine zugehörige Task-Mapping kann als Task-Mapping-Matrix berechnet werden. Die Task-Mapping-Matrix kann auch als mapping matrix (T) oder mapping matrix (U) bezeichnet werden. Die in Gleichung (12) beschriebene Matrix T_k, welche als „dynamisch konsistente Pseudo-Inverse“ der Transponierten der Task-Jacobian Matrix J_k bezeichnet werden kann, stellt ein Beispiel für eine solche Task-Mapping-Matrix dar. Die einzelnen Task-Mapping-Matrizen können vorteilhafterweise auch, wie in Gleichung (22) beschrieben, in einer vereinigten Task-Mapping-Matrix T kombiniert werden.

[0019] Wenigstens ein weiteres Reglermodul kann in den Gesamtregler integriert werden. Wenigstens ein als Zwangsbedingung formuliertes weiteres Reglermodul kann in den Gesamtregler integriert werden.

[0020] Das wenigstens eine erste Reglermodul wird gewichtet. Das wenigstens eine weitere Reglermodul kann gewichtet werden. Zur Gewichtung wird eine Gewichtungs-Matrix verwendet. Die Gewichtungs-Matrix kann auch als weighting matrix (W) bezeichnet werden.

[0021] Das wenigstens eine erste Reglermodul und/oder das wenigstens eine weitere Reglermodul können/kann derart gewichtet werden, dass der Gesamtregler auch bei einem überbestimmten Regelungsproblem ein zumindest annähernd passives Verhalten aufweist.

[0022] Das wenigstens eine erste Reglermodul und/oder das wenigstens eine weitere Reglermodul können/kann mithilfe wenigstens einer Pseudo-Inversen und/oder wenigstens einer Inversen optimiert werden.

[0023] Das wenigstens eine erste Reglermodul und/oder das wenigstens eine weitere Reglermodul können/kann mithilfe wenigstens einer Optimierungsvariablen optimiert werden. Das wenigstens eine erste Reglermodul und/oder das wenigstens eine weitere Reglermodul können/kann quadratisch optimiert werden. Die quadratische Optimierung kann durch eine einzige Instanz oder durch mehrere Instanzen ausgeführt werden.

[0024] Wenigstens eines der Reglermodule kann als Tracking-Regler ausgeführt werden. Wenigstens eines der Reglermodule kann als Regulation-Regler ausgeführt werden. Jedem regelbaren Freiheitsgrad kann ein gesondertes Reglermodul zugeordnet werden.

[0025] Das Computerprogrammprodukt kann auf einem computerlesbaren Speichermedium, auf einem computerlesbaren Datenträger oder als Datenträgersignal vorliegen.

[0026] Das Verfahren kann folgende Schritte aufweisen: a) Berechnung der gewünschten Task-Kräfte sowie der zugehörigen Task-Mapping Matrizen für sämtliche passiven Tracking-Regler-Module, wobei das nominelle Verhalten der passiven Tracking-Regler-Module dem von Feder-Masse-Dämpfer Systemen entspricht, wobei weiterhin die natürliche Trägheit des Roboters erhalten wird und wobei weiterhin die task-spezifischen Coriolis- und Zentrifugal-Effekte unberührt bleiben, während alle übrigen Coriolis- und Zentrifugal-Effekte kompensiert werden; b) Berechnung der speziellen passivitätsgarantierenden bzw. passivitäts-fördernden Gewichtungsmatrizen für sämtliche Tasks, c) Integration und Kombination der einzelnen passiven Tracking-Regler-Module durch Optimierung zu einem Gesamtregler unter Verwendung der speziellen passivitätsgarantierenden bzw. passivitäts-fördernden Gewichtungs-matrizen.

[0027] In Verfahrensschritt c) können zusätzlich zu den passiven Tracking-Regler-Modulen auch andere Regler bzw. Reglermodule integriert werden. In Verfahrensschritt c) kann/können zusätzlich zu den gewichteten Tasks auch ein oder mehrere Tasks als Zwangsbedingung formuliert und entsprechend durch die Optimierung in den Gesamt-Regler integriert werden. In Verfahrensschritt c) kann eine Pseudo-Inverse für die Optimierung verwendet werden. In Verfahrensschritt c) kann die Optimierung auf dem Konzept der Quadratischen Optimierung bzw. des „Quadratic Programming“ (QP) beruhen.

[0028] Die Erfindung kann ein Regler-Modul zur Regelung von drehmoment-basierten Robotern betreffen. Das Regler-Modul kann eine Task-Mapping-Matrix sowie einen gewünschten Task-Vektor aufweisen, wobei ein nominelles Verhalten des Regler-Moduls einem Feder-Masse-Dämpfer System entsprechen kann.

[0029] Die entsprechende Masse bzw. Trägheit der natürlichen Masse bzw. Trägheit des Roboters, gegebenenfalls projiziert in den entsprechenden Taskspace, kann, was als Compliance-Verhalten bezeichnet wird, also einem Impedanzverhalten entsprechen, welches auf der natürlichen Roboter-Masse bzw. der natürlichen Roboter-Trägheit basiert und somit auf das sogenannte Trägheits-Shaping verzichtet.

[0030] Die task-spezifischen Coriolis- und Zentrifugal-Effekte können unberührt bleiben, während alle übrigen Coriolis- und Zentrifugal-Effekte kompensiert werden, wodurch die Passivität des Regler-Moduls erzielt werden kann.

[0031] Das Regler-Modul kann als Tracking-Regler ausgeführt sein. Das Regler-Modul kann als Regulation-Regler ausgeführt sein und somit auf eine Vorsteuerung verzichten.

[0032] Einer beliebige Anzahl von derartige Regler-Modulen kann zu einem Gesamt-Regler kombiniert sein. Eine beliebige Anzahl von derartigen Regler-Modulen sowie eine beliebige Anzahl anderer Regler bzw. Regler-Module kann zu einem Gesamt-Regler kombiniert sein.

[0033] Die Erfindung kann einen derartigen Gesamt-Regler betreffen, wobei eine sanfte oder softe Priorisierung mittels Gewichtung der Tasks, entsprechend der Regler-Module, erfolgen kann.

[0034] Die Gewichtung der den Regler-Modulen entsprechenden Tasks kann derart ausgestaltet sein, dass die Passivität des Gesamt-Reglers garantiert wird bzw. ein Verhalten des Gesamt-Reglers erzeugt wird, welches bestmöglich an ein passives Verhalten herankommt, selbst wenn sich die Zielsetzungen der einzelnen Tasks widersprechen und somit ein überbestimmtes Regelungsproblem vorliegt.

[0035] Zusätzlich kann/können ein Task oder mehrere Tasks als harte Zwangsbedingung formuliert sein, während alle übrigen Tasks auf derartigen Regler-Modulen basieren und durch eine sanfte Gewichtung berücksichtigt werden bzw. in den Gesamt-Regler integriert werden.

[0036] Zur Lösung des Regelungsproblems kann eine Quadratische Optimierung zum Einsatz kommen, was sowohl durch eine einzige Instanz zur Quadratischen Optimierung als auch durch mehrere Instanzen zur Quadratischen Optimierung ausgeführt sein kann.

[0037] Zur Lösung des Regelungsproblems kann/-können eine oder mehrere Pseudo-Inverse oder Inverse zum Einsatz kommen.

[0038] Jedem regelbaren Freiheitsgrad kann ein eigenes Regler-Modul zugeordnet werden, wodurch eine maximale Granularität bzw. die maximal mögliche Entkopplung der einzelnen Task-Regler erreicht wird.

[0039] Zusammenfassend und mit anderen Worten dargestellt ergibt sich somit durch die Erfindung unter anderem ein modularer passiver Tracking-Regler für kraft/drehmomentgeregelte Roboter.

[0040] Der erfindungsgemäße Regler erlaubt eine präzise und gleichzeitig nachgiebige, sowie robuste Regelung von redundanten Robotersystemen. Dies wird erreicht durch den Einsatz von passivitätsbasierten Regler-Modulen, welche mittels verschiedener Optimierungsverfahren (in Abhängigkeit von der vorliegenden Problemstellung) zu einem Gesamtregler kombiniert werden. Durch den Einsatz einer speziellen Gewichtung der verschiedenen Tasks wird die Passivität des Gesamtreglers, wiederum in Abhängigkeit von der vorliegenden Problemstellung, entweder garantiert oder zumindest gefördert. Diese spezielle Gewichtung beruht darauf, dass für die einzelnen Tasks die jeweilige invertierte Task-Trägheitsmatrix multipliziert mit einer skalaren Task-Gewichtungszahl als Gewicht zum Einsatz kommt. Reglerentwurf und -analyse beruhen auf dem Konzept der Passivität bzw. auf Lyapunov Theorie, wodurch die hohe Robustheit der Regelung erzielt wird.

[0041] Das vorliegende Verfahren erweist sich als robust gegenüber Singularitäten, robust gegenüber Modellierungsfehlern und Messrauschen, sehr flexibel einsetzbar (Änderung der Reglerstruktur ohne weiteres online möglich, Kombination mit anderen Reglermethoden ohne weiteres möglich), sowie als sehr robust beim Kontakt mit der Umgebung. Der Tuningaufwand während realen Experimenten bzw. für reale Anwendungen ist sehr gering. Es können beliebig viele Tasks miteinander kombiniert werden; jeder Task trägt entsprechend ihrer Gewichtung zum Gesamtergebnis bei.

[0042] Somit vereint das vorliegende Verfahren die wichtigsten Vorteile aller im Stand der Technik beschriebenen Verfahren für die Regelung von redundanten, drehmomentgeregelten Robotersystemen, während sämtliche Nachteile vermieden werden.

[0043] Das vorliegende Verfahren deckt nahezu das gesamte Spektrum zwischen Inverse Dynamics basierten Reglern und den sehr robusten, sogenannten PD+ basierten Reglern ab. Die Granularität der gewünschten Entkopplung bzw. die Modularität der einzelnen Sub-Regler ist frei wählbar.

[0044] Das technische Gebiet, welches durch die vorliegende Erfindung insbesondere adressiert wird, ist die Regelung von redundanten, drehmomentgeregelten Robotersystemen. Hierbei stellt die Lösung von Multitasking-Aufgabenstellungen ein schwieriges Problem dar, insbesondere im Hinblick auf Stabilität und Passivität des Gesamtreglers, präzises Tracking von gewünschten Referenztrajektorien, Robustheit gegenüber Modellungenauigkeiten und Sensorrauschen, Robustheit gegenüber Singularitäten, Flexibilität der Implementierung und Kombinierbarkeit unterschiedlicher Teil-Regler, sowie Kontaktstabilität.

[0045] Der Stand der Technik umfasst im Wesentlichen zwei verschiedene Methoden, welche diese Aufgabenstellung teilweise lösen: zum einen Inverse Dynamics-basierte Verfahren, zum anderen passivitäts- bzw. impedanz-basierte Verfahren. Beide im Stand der Technik enthaltenen Lösungsansätze haben gewisse bisher nicht überwundene Nachteile:

> Inverse Dynamics ist nicht passiv, so dass in realen Einsatzszenarien schwerwiegende Stabilitätsprobleme auftreten können. Inverse Dynamics ist weiterhin nicht robust gegenüber Modellungenauigkeiten und Sensorrauschen, und außerdem anfällig gegenüber Singularitäten. Auch die Kontaktstabilität kann durch Inverse Dynamics-basierte Regelungsverfahren nicht immer gewährleistet werden.

[0046] Die aus dem Stand der Technik bekannten passivitäts- bzw. impedanz-basierten Verfahren weisen zumeist entweder ein schlechtes Tracking der gewünschten Referenztrajektorien oder eine Anfälligkeit gegenüber Singularitäten auf. Weiterhin ist die entsprechende Regler-Implementierung zumeist unflexibel und eine Kombination unterschiedlicher Teil-Regler bzw. Regler-Typen ist nicht vorgesehen.

[0047] Die vorliegende Erfindung beruht auf einem Verfahren zur Regelung von redundanten, drehmomentgeregelten Robotersystemen, in welchem passive bzw. impedanz-basierte Tracking Regler-Module durch eine Optimierung miteinander kombiniert werden. Durch eine spezielle Art der Gewichtung wird hierbei sichergestellt, dass durch die Kombination/Synthese der einzelnen Regler-Module die Passivität des Gesamtsystems nicht verloren geht bzw. dass durch die entsprechende Kombination/Synthese keine Aktivität, also eine entsprechende ungewollte Zunahme der Energie im System, bewirkt wird. Hierdurch werden sämtliche vorgenannten Vorteile erreicht und die entsprechenden Nachteile umgangen. Insbesondere stellt das der vorliegenden Erfindung zugrunde liegende Verfahren die Stabilität und Passivität des Gesamtsystems sicher bzw. för-

dert letztere im Falle, dass eine Stabilisierung bzw. Passivierung des Gesamtsystems technisch unmöglich ist, es erzielt präzises Tracking der gewünschten Referenztrajektorien, sofern diese nicht inkonsistent und somit technisch realisierbar sind, es ist robust gegenüber Singularitäten, Modellungenauigkeiten und Sensorrauschen, es zeichnet sich durch eine gute Kontaktstabilität aus, und es ermöglicht eine flexible Implementierung, in welcher unter anderem auch unterschiedliche Regler-Typen miteinander kombiniert werden können. Somit löst das der vorliegenden Erfindung zugrunde liegende Verfahren sämtliche hier adressierten Problemstellungen vollständig.

[0048] Nachfolgend werden Ausführungsbeispiele der Erfindung unter Bezugnahme auf Figuren näher beschrieben, dabei zeigen schematisch und beispielhaft:

Fig. 1 MPTC deckt den gesamten Bereich zwischen Inverse Dynamics (ID) und PD+ Regelung ab,

Fig. 2 Sprungantworten für ein voll bestimmtes, nicht widersprüchliches Task-Setup. Links: Gelenkwinkel und Referenzen. Rechts: gesamte und task-spezifische Lyapunov-Funktionswerte,

Fig. 3 Sprungantwort für überbestimmtes (und damit widersprüchliches) Task-Setup,

Fig. 4 der humanoide Roboter TORO geht in einer OpenHRP Simulation (Single- und Double-Support-Zeiten: TSS= 0.8s, TDS= 0.12s). Nach 2.5s wird der linke Fuß für 0.3s durch eine externe Kraft von -60N in x-Richtung gestört,

Fig. 5 TORO geht im Experiment (Single- und Double-Support-Zeiten: TSS= 0.9s, TDS= 0.3s, Schrittlänge 0.15m) und

Fig. 6 eine mögliche Anwendung des erfindungsgemäßen Verfahrens.

[0049] Beschrieben wird eine Ausführung eines Modularen Passiven Tracking Controllers (MPTC) für „Stack of Tasks“-basierte Regler-Frameworks.

Abstract

[0050] Diese Arbeit führt den sogenannten Modularen Passiven Tracking Controller (MPTC) ein, welcher ein generischer passivitäts-basierter Regler ist, der darauf abzielt, die Ziele verschiedener Unteraufgaben (Subtasks) unabhängig voneinander zu erfüllen. Diese werden in einem Stack of Tasks (SoT) miteinander kombiniert, welcher als Basis für die Synthese eines Gesamtsystem-Reglers dient. Die entsprechende Analyse und das Regler-Design basieren auf Lyapunov-Theorie. Ein wichtiger Beitrag dieser Arbeit ist das Design einer spezifischen Optimierungs-Gewichtungs-Matrix, welche die Passivität eines überbestimmten und damit widersprüchlichen Task-Setups sicherstellt. Das vorgeschlagene Framework wird durch Simulationen und Experimente validiert, sowohl für fest montierte als auch für free-floating Roboter.

I. EINLEITUNG

[0051] Die simultane Regelung verschiedener Aufgaben hat sich zu einem bedeutenden Forschungsthema auf dem Gebiet der Roboterregelung entwickelt. Während anfängliche Arbeiten den einfacheren Fall einer einzigen Aufgabe und ihres Nullraumes für kinematisch redundante Roboter betrachteten, existieren heutzutage mehrere etablierte Frameworks zur Behandlung multipler Aufgaben mit und ohne Prioritäten. In der Literatur wird zwischen Arbeiten unterschieden, welche das Aufgaben-KoordinationsProblem zunächst auf kinematischer Ebene lösen, und Arbeiten, welche die Regelung direkt für die Dynamik formulieren. Eine weitere wichtige Klassifizierung kann basierend auf der Nutzung strikter Aufgaben-Prioritäten mittels hierarchischer Regler im Vergleich zu Reglern, welche eine weiche Priorisierung mittels Aufgabengewichtung anwenden, durchgeführt werden.

[0052] Im Bereich der kinematischen Regelung wurden hierarchische Regler vorgeschlagen, welche auf sukzessiven oder augmentierten Nullraum-Projektionen beruhen, um eine strikte Aufgaben-Hierarchie [18, 2] sicherzustellen. Zur Behandlung von Aufgaben-Singularitäten wurde eine singularitätsrobuste Inverskinematik vorgeschlagen [4]. Allerdings zerstört diese singularitätsrobuste Inverse die strikte Aufgaben-Hierarchie und generiert im Endeffekt eine Gewichtung zwischen den verschiedenen Aufgaben.

[0053] Andere Frameworks behandeln multiple Aufgaben auf der Dynamik-Ebene. Der Operational Space Ansatz wurde durch Anwendungen in der humanoiden Robotik [22, 23] in diese Richtung erweitert. Andere Inverse Dynamics (ID) basierte Regler benutzen hierarchische Quadratische Programme (QP) [20, 11, 3]. Die meisten dieser Arbeiten zielen auf eine strikte Aufgabenentkopplung ab.

[0054] Die vorliegende Arbeit ist inspiriert durch die Familie von Inverse Dynamics basierten Trackingreglern, welche mehrere Aufgaben (welche in einem Stack of Tasks (SoT) kombiniert werden) mithilfe eines einzigen gewichteten QP's [13, 15, 10] weich gegeneinander abwägen. Derartige Regler sind einfach zu schreiben und zeichnen sich durch ihre hohe Flexibilität aus. Diese Regler sind allerdings im Vergleich zu passivitäts-basierten Ansätzen wie beispielsweise [19, 5, 12, 16, 6] weniger robust

gegenüber Modellierungsfehlern und Kontaktunsicherheiten [10, 6]. Dies bewirkt bei der Umsetzung auf realen Systemen Probleme wie beispielsweise Vibrationen, welche oft durch heuristische Ansätze adressiert werden [13, 10].

[0055] Des weiteren dienten der gewichtungsbasierte multi-objective Regler in [3] und der strikt hierarchische passivitätsbasierte Regler aus [6] als Inspiration für diese Arbeit. Ähnlich wie in [3] benutzen wir hier ein QP, um individuelle Regelziele aus unterschiedlichen Task-Space-Reglern zu kombinieren. In [3] wird allerdings jedes einzelne Regelziel basierend auf Inverse Dynamics mit der Einheitsmatrix als gewünschter Trägheit (Feedback Linearisierung) berechnet. Im Gegensatz dazu benutzen die individuellen Task-Regler, welche hier vorgestellt werden, das Konzept der Passivität und vermeiden eine Modifikation der Trägheit, das heißt, wir zielen auf ein PD+ artiges Closed-Loop-Verhalten für jede einzelne Task [19] ab. Im Vergleich zu [6], worin ebenfalls die natürliche Trägheit erhalten wird, benutzen wir eine gewichtete QP Formulierung (weiche Priorisierung), was uns erlaubt, eine beliebige Anzahl von verschiedenen Tasks miteinander zu kombinieren, und was weiterhin in gewissen Situationen (beispielsweise wenn eine einzelne Task singulär wird) zu einem weniger aggressiven Verhalten führt.

[0056] In dieser Arbeit leiten wir eine Regelungs-Architektur her, welche auf nominell passiven Subtask-Reglern basiert, den sogenannten Modularen Passiven Tracking Controllern (MPTC). Diese werden mit Hilfe eines Stack of Tasks miteinander kombiniert und gegeneinander aufgewogen, wobei der Stack of Tasks durch eine einzelne gewichtete Pseudo-Inverse oder ein entsprechendes QP gelöst wird. Das Regelungs-Framework kombiniert die Vorteile sowohl von Inverse Dynamics Reglern als auch von passivitäts-basierten Reglern, nämlich: Einfachheit der Implementierung und Anwendung, die Unterstützung von Task Space Tracking, Passivität und Kontakt-Robustheit, und die natürliche Bewältigung von Singularitäten. Die entsprechende Stabilitätsanalyse basiert auf Lyapunov-Theorie. Es zeigt sich, dass der Gesamt-Regler für den Fall, dass sich die Aufgaben nicht widersprechen, asymptotisch stabil und passiv ist. Ein wichtiger Beitrag der vorgestellten Arbeit ist die Herleitung eines spezifischen Optimierungs-Gewichtes, welches zusätzlich die Passivität sogar im überbestimmten (das heißt widersprüchlichen) Fall erhält. Für konkurrierende Tasks und entsprechende inkonsistente Task-Referenzen zeigen mehrere Simulationen die Stabilität und Robustheit von MPTC sogar im Tracking-Fall, wobei ein formaler Stabilitätsbeweis derzeit noch fehlt.

[0057] Das Paper ist folgendermaßen aufgebaut: Kapitel II leitet den Modularen Passiven Tracking Controller (MPTC) auf Task-Ebene her, während Kapitel III die entsprechende Analyse der Gesamt-Closed-Loop und die Regler-Herleitung vorstellt. Kapitel IV vergleicht MPTC mit Inverse Dynamics (ID) und PD+ basierten Reglern, und präsentiert eine große Auswahl an möglichen Ausprägungen der Entkopplung. Kapitel V präsentiert Simulationsergebnisse sowohl für fixed-base als auch für free-floating Roboter, während Kapitel VI das Paper beschließt.

II. HERLEITUNG DES MODULAREN PASSIVEN TRACKING CONTROLLERS (MPTC)

[0058] Diese Arbeit betrachtet n_T Aufgaben (Tasks), wobei jede davon ihr eigenes individuelles Ziel hat. Um die Ziele der einzelnen Tasks zu befriedigen, leitet dieses Kapitel Modulare Passive Tracking Controller (MPTC) her, welche in Kapitel III zu verschiedenen Gesamtreglern kombiniert werden.

A. Allgemeines Robotermodell

[0059] Die allgemeine Bewegungsgleichung eines Roboters kann geschrieben werden als

$$M(q)\ddot{q} + C(q,\dot{q})\dot{q} + \tau_s(q) = \tau, \qquad (1)$$

wobei $q \in \mathbb{R}^n$ die generalisierten Koordinaten (Im Falle eines fixed-base Roboters bestehen diese einfach aus den Gelenk-Koordinaten q_j (das heißt $q = q_j$), während sie im Falle von free-floating Robotern (wie beispielsweise Humanoiden) zusätzlich die Roboter-Basis-Koordinaten x enthalten (das heißt $q = \left[x_b^T, q_j^T\right]^T$) bezeichnet, M(q), C(q,q̇) und $\tau_g(q)$ kennzeichnen die Trägheitsmatrix, Coriolis- und Zentrifugal-Matrix, und die Gravitations-Drehmomente (zu beachten: die Abhängigkeiten von q und q̇ werden im Folgenden vernachlässigt.), und

$$\tau = S^T\left(\tau_j + \tau_{ini}\right) + \underbrace{L_{all}^T w_{all}}_{\tau_{ext}} \qquad (2)$$

bezeichnet die generalisierten Kräfte. Diese setzen sich einerseits aus Gelenks-Motordrehmomenten τ_j und internen Störmomenten τ_{int}, welche in den Robotergelenken wirken (z.B. Gelenkreibung), zusammen, welche beide über die Gelenks-Auswahl-Matrix S (Zu beachten: für fixed-base Roboter ist S eine Einheitsmatrix, während für free-floating Roboter $S = \left[0_{n_{act}\times\sigma}, I_{n_{act}\times n_{act}}\right]$ gilt, wobei n_{acl} die Anzahl der aktuierten Robotergelenke bezeichnet.) auf τ wirken, und andererseits aus externen Drehmomenten τ_{ext}. Letztere setzen sich aus der Gesamtheit aller Wrenches (Kombination aus Kräften und Drehmomenten) $w_{all} = \left[w_1^T, \ldots, w_{n_1}^T\right]^T$, welche auf die n_L

Roboter-Glieder (Links) wirken, und über die gestapelten Link-Jacobians $L_{all} = \left[L_1^T, \ldots, L_{n_t}^T \right]^T$ auf τ wirken. Während Kapitel III-E die einzelnen Elemente von (2) benutzt, werden wir im Folgenden einfach τ benutzen, um beliebige generalisierte Kräfte zu repräsentieren. Durch Auflösen von (1) nach den generalisierten Beschleunigungen $\ddot{q}$ erhalten wir

$$\ddot{q} = M^{-1}\left(\tau - C\ \dot{q} - \tau_g\right), \qquad (3)$$

B. Task Space Größen

I) Task Space Geschwindigkeiten und Beschleunigungen: In der Roboter-Regelung kann für typische Task Spaces (Typische Tasks (Aufgaben) sind beispielsweise Gelenkregler und kartesische Endeffektor-Regler) ein Task Geschwindigkeits-Vektor

$$\dot{x}_k = J_k \dot{q} \qquad (4)$$

formuliert werden. Der Index k zeigt an, dass ein solches Mapping für alle n_T Tasks existiert, das heißt $k \in \{1,.., n_T\}$, Hierbei bezeichnet $J_k \in \mathbb{R}^{n_{t_k} \times n}$ die entsprechende Task-Jacobian, mit der Dimensionalität $n_{T,k}$ der k-ten Task. Durch Ableiten von (4) bezüglich der Zeit und Einsetzen von (3) finden wir die Task Space Beschleunigung

$$\ddot{x}_k = \dot{J}_k \dot{q} + J_k \ddot{q} = J_k M^{-1}\left(\tau - \tau_g\right) - Q_k \dot{q}, \qquad (5)$$

wobei

$$Q_k = J_k M^{-1} C - \dot{J}_k, \qquad (6)$$

[0060] Für das Design eines Tracking-Reglers (siehe Kapitel II-C) sind der entsprechende Task-Geschwindigkeitsfehler

$$\dot{\tilde{x}}_k = \dot{x}_{k,ref} - \underbrace{J_k\ \dot{q}}_{\dot{x}_k} \qquad (7)$$

und der Task-Beschleunigungsfehler

$$\ddot{\tilde{x}}_k = \ddot{x}_{k,ref} - \underbrace{\left(J_k\ M^{-1}\left(\tau - \tau_g\right) - Q_k \dot{q}\right)}_{\ddot{x}_k} \qquad (8)$$

von besonderem Interesse. Hierbei bezeichnen $\dot{x}_{k,ref}$ und $\ddot{x}_{k,ref}$ die Task-Referenzgeschwindigkeit und Task-Referenzbeschleunigung.

2) Task Space Trägkeit und ihre Ableitung: Unter Verwendung von J_k kann die Trägheitsmatrix M in den Task Space [17] projiziert werden:

$$M_k = \left(J_k\ M^{-1}\ J_k^T\right)^{-1}. \qquad (9)$$

[0061] Ableiten von (9) ergibt

$$\begin{aligned} \dot{M}_k &= -M_k \Big(\dot{J}_k M^{-1} J_k^T - J_k M^{-1} \overbrace{\dot{M}}^{C+C^T} M^{-1} J_k^T \\ &\qquad + J_k M^{-1} \dot{J}_k^T \Big) M_k \\ &= C_k + C_k^T, \end{aligned} \qquad (10)$$

mit der Task Space Coriolis- und Zentrifugal-Matrix

$$C_k = M_k Q_k T_k^T. \qquad (11)$$

[0062] Die Matrix

$$T_k = M_k J_k M^{-1} \qquad (12)$$

bezeichnet die dynamisch konsistente Pseudo-Inverse von J_k^T.

C. Modularer Passiver Tracking Controller (MPTC)

[0063] Dieses Kapitel leitet den vorgeschlagenen Modularen Passiven Tracking Controller (MPTC) her. Dieser wird in generischer Form geschrieben, wodurch er als Vorlage für beliebige spezifische Regler (wie beispielsweise kartesische oder Gelenk-Regler) dient. Für jede einzelne der n_T Tasks benutzen wir eine separate Lyapunov-Funktion basierend auf der task-bezogenen relativen kinetischen Energie $E_{kin,k}$ und der relativen potentiellen Energie $E_{pot,k}$:

$$V_k = \underbrace{\frac{1}{2} \dot{\tilde{x}}_k^T M_k \dot{\tilde{x}}_k}_{E_{kin,k}} + \underbrace{\frac{1}{2} \tilde{x}_k^T K_k \tilde{x}_k}_{E_{pot,k}} \qquad (13)$$

wobei die positiv definite, symmetrische Matrix K_k die Task-Steifigkeit bezeichnet. Diese Lyapunov-Funktion ist positiv definit bezüglich Task-Positionsfehler $\tilde{x}_k$ und Task-Geschwindigkeitsfehler $\dot{\tilde{x}}_k$.

[0064] Durch Ableiten von (13) und Einsetzen von (8) ergibt sich:

$$\begin{aligned} \dot{V}_k &= \dot{\tilde{x}}_k^T \left(M_k \ddot{\tilde{x}}_k + \frac{\dot{M}_k}{2} \dot{\tilde{x}}_k + K_k \tilde{x}_k \right) \\ &= \dot{\tilde{x}}_k^T \Bigg(\underbrace{M_k J_k M^{-1}}_{T_k} \left(\tau_g - \tau\right) + M_k Q_k \dot{q} \\ &\qquad + M_k \ddot{x}_{k,ref} + C_k \dot{\tilde{x}}_k + K_k\ \tilde{x}_k \Bigg). \end{aligned} \qquad (14)$$

[0065] Hierbei wurde folgende Gleichung benutzt

$$\frac{\dot{M}_k}{2} = \frac{C_k^T + C_k}{2} = \underbrace{\frac{C_k^T - C_k}{2}}_{\text{skew-symmetric}} + C_k, \qquad (15)$$

welche es uns erlaubt, $\dot{\tilde{x}}_k^T \frac{\dot{M}_k}{2} \dot{\tilde{x}}_k$ als $\dot{\tilde{x}}_k^T C_k \dot{\tilde{x}}_k$ umzuschreiben, da der schiefsymmetrische Term herausfällt.

[0066] Jetzt definieren wir die (tatsächliche) Task-Kraft (Zu beachten: in Abhängigkeit von der Aufgabe kann die Task-Kraft lineare Kräfte, Drehmomente, Wrenches etc. enthalten.) f_k als:

$$f_k = T_k \tau. \qquad (16)$$

[0067] Durch die Wahl der gewünschten Task-Kraft (Zu beachten: diese gewünschte Task-Kraft $f_{k,des}$ wird in Kapitel III als task-spezifisches Regler-Ziel verwendet werden. Ebenfalls zu beachten: In Kapitel IV-B, stellen wir eine alternative (jedoch äquivalente) Reglerformulierung (55) vor, welche die Ähnlichkeit des Reglers zu einer PD+ Formulierung leichter erkennen lässt.) $f_{k,des}$ als

$$f_{k,des} = T_k \tau_g + M_k Q_k \dot{q} + M_k \ddot{x}_{k,ref} + (C_k + D_k)\dot{\tilde{x}} + K_k \tilde{x}_k \qquad (17)$$

und durch Umschreiben von $T_k \tau$ als

$$T_k \tau = f_k = f_{k,des} - \underbrace{\left(f_{k,des} - f_k\right)}_{\tilde{f}_k}, \qquad (18)$$

wird die Einzel-Task-Lyapunov-Rate aus (14) zu

$$\dot{V}_k = \underbrace{-\dot{\tilde{x}}_k^T D_k \dot{\tilde{x}}_k}_{\dot{V}_{k,des},\text{purely dissipative}} + \underbrace{\dot{\tilde{x}}_k^T \tilde{f}_k}_{\dot{\tilde{V}}_k}. \qquad (19)$$

[0068] Man beachte, dass das geregelte System (auf Task-Ebene) passiv bezüglich des Einganges $\tilde{f}_k$, des Ausganges $\dot{\tilde{x}}_k$ und der Speicherfunktion V_k aus (13) ist. Während die gewünschte Lyapunov-Rate $\dot{V}_{k,des}$ für eine positiv definite Dämpfungs-Matrix D_k rein passiv ist, kann der Term $\dot{\tilde{V}}_k$ ungleich Null sein, in Abhängigkeit von Faktoren wie unbekannte Störungen, Unteraktuierung und andere Aktuatorlimits, Task-Inkonsistenzen und Priorisierung. Schließlich multiplizieren wir (8) mit M_k vor, setzen (18) und (17) ein, vereinfachen und stellen um, um die Task-Dynamik in folgender Form darzustellen

$$M_k \ddot{\tilde{x}}_k + (C_k + D_k)\dot{\tilde{x}}_k + K_k \tilde{x}_k = \tilde{f}_k. \qquad (20)$$

[0069] Man beachte, dass der task-bezogene Coriolis-Term (das heißt $C_k \dot{\tilde{x}}_k$) nicht gekürzt wurde, was eine Voraussetzung für Passivität darstellt. Wenn die gewünschte Task-Kraft erreicht wird (das heißt $\tilde{f}_k = 0$) entspricht Gleichung (20) einer Feder-Masse-Dämpfer Dynamik für Task k. Im Falle von kompatiblen Tasks kann asymptotische Stabilität aller Trajektorien gezeigt werden. Dies kann beispielsweise erreicht werden, indem die ε-Methode [17] verwendet wird, um eine starke Lyapunov Funktion mit negativ definiter Zeitableitung, ähnlich wie in [6], zu erhalten.

[0070] Andernfalls, zum Beispiel im Falle von Unteraktuierung oder anderen Aktuator-Limits, unerwarteten externen Störungen oder widersprüchlichen Tasks, entspricht (20) einem Compliance (Zur Erinnerung: Compliance bedeutet ein Impedanz-Verhalten mit natürlicher Trägheit (keine Modifikation der Trägheit).) Verhalten [7]. Für solche Fälle können weiterführende Analysen notwendig werden. In dieser Arbeit konzentrieren wir uns auf das Problem der Task-Inkonsistenzen, welche in den folgenden Kapiteln adressiert werden.

III. ANALYSE DES CLOSED-LOOP VERHALTENS DES GESAMTSYSTEMS UND GESAMT-REGLER

[0071] In diesem Kapitel leiten wir Regler für das Gesamtsystem, bestehend aus den Subtask-Reglern, her, das heißt für das komplette Set von gewünschten Task-Kräften $f_{k,des}$ aus (17) für $k \in \{1,..,n_T\}$, und wir analysieren deren Closed-Loop Verhalten durch die Anwendung der Lyapunov-Theorie.

A. Definition von verschiedenen Task-Kraft-Fehlern

[0072] Durch Kombination aller gewünschten Task-Kräfte $f_{k,des}$ aus (17) für $k \in \{1,.., .., n_T\}$ erhalten wir

$$f_{des} = \begin{bmatrix} f_{1,des} \\ \vdots \\ f_{n_\tau,des} \end{bmatrix}. \qquad (21)$$

[0073] In ähnlicher Weise kombinieren wir auch (16) für alle $k \in \{1, .., n_T\}$ und erhalten

$$\underbrace{\begin{bmatrix} f_1 \\ \vdots \\ f_{n_\tau} \end{bmatrix}}_{f} = \underbrace{\begin{bmatrix} T_1 \\ \vdots \\ T_{n_\tau} \end{bmatrix}}_{T} \tau. \qquad (22)$$

[0074] Dies stellt das Mapping der tatsächlichen generalisierten Kräfte τ auf den Stack (Stapel) der tatsächlichen Task-Kräfte über die entsprechende vereinigte Task-Mapping-Matrix $T \in \mathbb{R}^{n_{T,all} \times n}$ dar, wobei die Summe über alle Subtask-Dimensionalitäten durch $n_{T,all} = \sum_{k=1}^{n_T} n_{T,k}$ bezeichnet wird. Nun definieren wir den Stack der tatsächlichen Task-Kraft Fehler

$$\tilde{f} = f_{des} - f = f_{des} - T\tau. \qquad (23)$$

[0075] Um die Diskussion über kommandierte Task-Kräfte und entsprechende Fehler zu vereinfachen, evaluieren wir nun (22) für die kommandierten generalisierten Kräfte τ_{cmd}, welche durch einen Regler (Mögliche Regler können zum Beispiel pseudo-inversen basierte Optimierung wie in (36) für unbeschränkte Regelungsprobleme benutzen oder Quadratic Programming (QP) basierte Optimierung (siehe Kapitel III-E), um Ungleichungs-basierte Nebenbedingungen zu behandeln.) generiert werden, und erhalten

$$f_{emd} = \begin{bmatrix} f_{1,emd} \\ \vdots \\ f_{n,emd} \end{bmatrix} = T\ \tau_{emd} = \underbrace{TU}_{T_u}\ u_{emd}, \qquad (24)$$

wobei f_{cmd} den Stack der kommandierten Task-Kräfte darstellt. Für bestimmte Regelungs-Probleme (wie beispielsweise Unteraktuierung, siehe Kapitel III-E) kann τ_{cmd} aus den gegebenen Optimierungsvariablen u_{cmd} resultieren, welche über die Aktuierungs-Mapping-Matrix U, also durch $\tau_{cmd} = U\ u_{cmd}$, auf die entsprechenden generalisierten Kräfte gemappt werden.

[0076] Indem wir (24) von (21) abziehen, erhalten wir schließlich den entsprechenden Stack der kommandierten Task-Kraft-Fehler:

$$\tilde{f}_{emd} = f_{des} - f_{emd} = f_{des} - T\ \tau_{emd} = f_{des} - T_u u_{emd}. \qquad (25)$$

B. Definition einer Lyapunov-Funktion für das Gesamtsystem und deren tatsächliche und kommandierte Ableitungen

[0077] In diesem Kapitel werden wir die Stabilität und Passivität von kompletten Sets aus modularen Task-Space-Reglern (und damit die Stabilität des robotischen Gesamtsystems, falls alle Roboterfreiheitsgrade durch die kombinierten Task-Koordinaten abgedeckt werden) analysieren. Hierfür kombinieren wir alle Einzel-Task Lyapunov-Funktionen (13), um die folgende Gesamt-Lyapunov-Funktion zu konstruieren

$$V = \sum_{k=1}^{n_T} (\psi_k V_k), \qquad (26)$$

welche für positive skalare Gewichte (Zu beachten: Diese positiven skalaren Gewichte ψ_k sind äquivalent zu den Optimierungsgewichten, welche in (34) bzw. (41) benutzt werden.) $\psi_k > 0$ positiv definit ist (genau wie ihre Eingangselemente V_k). In ähnlicher Weise erhalten wir die Gesamt-Lyapunov-Funktions-Ableitung $\dot{V}$ durch die Kombination der Lyapunov-Funktions-Ableitungen $\dot{V}_k$ der einzelnen Tasks aus (19):

$$\dot{V} = \sum_{k=1}^{n_T} \left(\psi_k \dot{V}_k\right) = \underbrace{\sum_{k=1}^{n_T} \left(\psi_k \dot{V}_{k,des}\right)}_{\dot{V}_{des}} + \underbrace{\sum_{k=1}^{n_T} \left(\psi_k \dot{\tilde{V}}_k\right)}_{\dot{\tilde{V}}}. \qquad (27)$$

[0078] Der Term $\dot{V}_{des}$ ist negativ semi-definit falls alle $\psi_k > 0$, da alle gewünschten Task-Lyapunov-Funktions-Ableitungen $\dot{V}_{k,des}$ negativ semi-definit sind. Der tatsächliche Fehler $\dot{\tilde{V}}$ der Gesamt-Lyapunov-Funktions-Ableitung kann folgendermaßen geschrieben werden

$$\dot{\tilde{V}} = \dot{\tilde{x}}^T \underbrace{\begin{bmatrix} \psi_1 I & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \psi_{n_T} I \end{bmatrix}}_{\Psi} \tilde{f}, \qquad (28)$$

wobei $\dot{\tilde{x}}^T = \left[\dot{\tilde{x}}_1^T, \ldots, \dot{\tilde{x}}_{n_T}^T\right]$. Man beachte, dass $\dot{\tilde{V}}$ $\dot{V}$ eine Funktion von f, und somit auch der tatsächlichen generalisierten Kräfte τ, ist.

[0079] Wir werden nun zusätzlich die Auswirkung des Stacks der kommandierten Task-Kraft-Fehler $\tilde{f}_{cmd}$ untersuchen. Durch entsprechende Abpassung von (28) finden wir den kommandierten Fehler der Gesamt-Lyapunov-Funktions-Ableitung

$$\dot{\tilde{V}}_{cmd} = \dot{\tilde{x}}^T \Psi \tilde{f}_{cmd}, \qquad (29)$$

welcher der folgenden kommandierten Gesamt-Lyapunov-Funktions-Ableitung entspricht

$$\dot{V}_{cmd} = \dot{V}_{des} + \underbrace{\sum_{k=1}^{n_T} \left(\psi_k \dot{\tilde{V}}_{k,cmd}\right)}_{\dot{\tilde{V}}_{cmd}}, \qquad (30)$$

wobei $\dot{\tilde{V}}_{k,cmd} = \dot{\tilde{x}}_k^T \tilde{f}_{k,cmd}$. Schließlich formulieren wir (28) folgendermaßen um

$$\begin{aligned} \dot{\tilde{V}} &= \dot{\tilde{x}}\ \Psi \left(\frac{\tilde{f}}{f_{des} - f} \right) \\ &= \underbrace{\dot{\tilde{x}}^T\ \Psi \left(\frac{\tilde{f}_{cmd}}{f_{des} - f_{cmd}} \right)}_{\dot{\tilde{V}}_{cmd}} + \underbrace{\dot{\tilde{x}}^T\ \Psi \left(\frac{\tilde{f}_{cmd}}{f_{cmd} - f} \right)}_{\dot{\tilde{V}}_{real}}, \end{aligned} \qquad (31)$$

was der tatsächlichen Gesamt-Lyapunov-Funktions-Ableitung

$$\dot{V} = \dot{V}_{des} + \dot{\tilde{V}}_{cmd} + \dot{\tilde{V}}_{real} \qquad (32)$$

entspricht, welche sich auf das reale Systemverhalten bezieht. Hierbei bezeichnet

$$\dot{\tilde{V}}_{real} = \dot{\tilde{x}}\,\Psi\left(\overbrace{\tilde{f}_{real}}^{\overbrace{f_{cmd}-f}}\right) \qquad (33)$$

die Komponente des Fehlers der Lyapunov-Funktions-Ableitung, welche den Abweichungen $\tilde{f}_{real}$ der kommandierten Task-Kräfte $\tilde{f}_{cmd}$ von den tatsächlichen Task-Kräften f entspricht.

C. Kostenfunktion für das Gesamtsystem

[0080] Basierend auf $\tilde{f}_{cmd}$ aus (25) formulieren wir die folgende Kostenfunktion für das Gesamtsystem

$$\begin{aligned} G &= \frac{1}{2}\tilde{f}_{cmd}^T W\,\tilde{f}_{cmd} \\ &= \frac{1}{2}\tau_{cmd}^T T^T W T \tau_{cmd} - f_{des}^T W T \tau_{cmd} + \frac{1}{2} f_{des}^T W f_{des} \\ &= \frac{1}{2} u_{cmd}^T T_u^T W T_u u_{cmd} - f_{des}^T W T_u u_{cmd} + \frac{1}{2} f_{des}^T W f_{des}. \end{aligned} \qquad (34)$$

[0081] Hierbei bezeichnet W eine beliebige symmetrische und positiv-definite Gewichtungsmatrix. Diese Kostenfunktion G wird durch die verschiedenen Regler, welche in den Kapiteln III-D und III-E präsentiert werden, minimiert.

D. Unbeschränkter und voll aktuierter Fall

[0082] I) Pseudo-Inversen-basierte allgemeine analytische Optimierung: In diesem Kapitel betrachten wir den voll aktuierten, unbeschränkten und potentiell widersprüchlichen Fall. Das Fehlen von ungleichungs-basierten Nebenbedingungen ermöglicht eine analytische Lösung durch gewichtete Pseudo-Inverse, wobei die volle Aktuierung die Regelbarkeit garantiert und es erlaubt, direkt die kommandierten generalisierten Kräfte τ_{cmd} als Optimierungsvariablen zu nutzen. In diesem Fall leiten wir G nach τ_{cmd}ab, um die Kostenfunktion (34, zweite Zeile) zu optimieren:

$$\frac{dG}{d\tau_{cmd}} = \tau_{cmd}^T T^T W\ T - f_{des}^T W\ T. \qquad (35)$$

[0083] Die Kostenfunktion (34) wird durch Null-Setzen von (35) und Auflösen nach den Regler-Kräften τ_{cmd} minimiert:

$$\tau_{cmd} = \left(T^T W\ T\right)^{-1} T^T\ W f_{des}, \qquad (36)$$

[0084] Dies sind die optimalen generlisierten Kraft-Kommandos für das gegebene Problem. Einsetzen von (36) in (25) ergibt

$$\tilde{f}_{cmd} = \underbrace{\left(I - T\left(T^T\ W\ T\right)^{-1} T^T\ W\right)}_{E_T} f_{des}, \qquad (37)$$

wobei E_T die Task-Kraft Ausgleichs-Matrix bezeichnet.

[0085] 2) nicht-widersprüchlicher Fall: Falls T quadratisch und invertierbar ist (d.h. alle Sub-Tasks sind unabhängig voneinander und widersprechen einander nicht), wird die Ausgleichs-Matrix aus (37) zu $E_T = 0$ und die kommandierten Task-Kraft-Fehler werden zu $\tilde{f}_{cmd} = 0$. In diesem Fall wird $\dot{\tilde{V}}_{cmd}$ aus (29) ebenfalls zu Null und somit wird $\dot{V}$ aus (32) zu

$$\dot{V} = \dot{V}_{des} + \dot{\tilde{V}}_{real}.$$

(38)

[0086] 3) Widersprüchlicher Fall: Falls T hingegen nicht invertierbar ist (z.B. im Falle von überbestimmten Regler-Setups), ist die Ausgleichsmatrix E_T ungleich Null. Durch Einsetzen von (37) in (29) finden wir den entsprechenden kommandierten Fehler der Gesamt-Lyapunov-Funktions-Ableitung als

$$\dot{\tilde{V}} = \left(\dot{x}_{ref}^T - \dot{q}^T\ J^T\right)\Psi\, E_T\, f_{des}. \qquad (39)$$

[0087] Hierbei bezeichnet $J = \left[J_1^T, ..., J_n^T,\right]^T$ den Stack der Einzel-Task-Jacobians.

[0088] Wir betrachten zunächst den widersprüchlichen Regula-tions-Fall (d.h. $\dot{x}_{ref} = 0$) und untersuchen die Matrix $J^T\,\Psi\,E_T$, welche in (39) erscheint:

$$\begin{aligned} J^T \Psi\ E_T &= J^T \Psi\left(I - T\left(T^T W T\right)^{-1} T^T W\right) \\ &= J^T \Psi\left(I - \Lambda J\left(J^T \Lambda W \Lambda J\right)^{-1} J^T \Lambda W\right). \end{aligned} \qquad (40)$$

[0089] Hier haben wir $T = \Lambda\, J\, M^{-1}$ eingesetzt, wobei Λ eine block-diagonale Matrix ist, welche die Task-Space-Trägheitsmatrizen $W = \Lambda^{-1}\Psi$ als diagonale Submatrizen enthält. Für eine beliebige Wahl von W (sogar wenn W als Diagonal-Matrix gewählt wird, wie dies oftmals in der Inverse Dynamics (ID) bezogenen Literatur zu finden ist) ist (40) ungleich Null, so dass $\left\{M_1, ..., M_{n_T}\right\}$ aus (39) ungleich Null ist, sogar im Regulations-Fall. Wenn allerdings

$$W = \Lambda^{-1}\Psi \qquad (41)$$

gewählt wird, wird (40) zu

$$J^T \Psi E_T = J^T \Psi - \underbrace{J^T \Psi \Lambda J \left(J^T \Psi \Lambda J \right)^{-1}}_{I} J^T \Psi = 0. \quad (42)$$

[0090] Dies bedeutet, dass (unabhängig von den aktuellen generalisierten Geschwindigkeiten $\dot{q}$ und gewünschten Task-Kräften f_{des}) für die Wahl (41) die kommandierte Lyapunov-Funktions-Ableitung $\tilde{V}_{cmd}$ aus (39) für den Regulations-Fall ($\dot{x}_{ref} = 0$) zu

$$\dot{\tilde{V}}_{cmd} = 0 \quad (43)$$

wird, und $\dot{V}$ aus (32) wird zu

$$\dot{V} = \dot{V}_{des} + \dot{\tilde{V}}_{real}. \quad (44)$$

[0091] Eine Analyse der Elemente von (44) ergibt, dass das geregelte Gesamtsystem passiv ist bezüglich des Einganges $\Psi\, \tilde{f}_{real,}$ des Ausganges $\dot{\tilde{x}}$ (wobei diese zwei die Elemente von $\dot{\tilde{V}}_{real}$ darstellen) und der positiv-definiten Speicherfunktion V aus (26). Entsprechend können wir auf Passivität des Gesamtsystems im hier untersuchten widersprüchlichen Regulations-Fall schließen. Für die Gesamtsystem-Closed-Loop-Dynamik von überbestimmten / widersprüchlichen Task-Setups (Regulations-Fall) fungiert (41) als passivitäts-garantierende Optimierungs-Gewichtung (PWOW). Man beachte, dass (43) nicht heißt, dass die Lyapunov-Funktions-Ableitungsfehler $\dot{\tilde{V}}_{k,cmd}$ der Einzel-Tasks zu Null werden (nur ihre Summe).

[0092] Die Gesamt-Gewichtungs-Matrix $W = \Lambda^{-1}\Psi$ aus (41) ist symmetrisch und block-diagonal. Ihre Symmetrie-Eigenschaft resultiert aus der Symmetrie ihrer diagonalen Sub-Matrizen

$$W_k = \psi_k M_k^{-1}, \quad (45)$$

welche eine Funktion von ψ_k und der Inversen der (symmetrischen) Task-Trägheitsmatrix M_k sind. Beim Betrachten von (45) wird klar, dass jede Task durch ihren entsprechenden Gewichtungs-Skalar ψ_k immer noch unabhängig (bezüglich der anderen Tasks) gewichtet werden kann.

[0093] Nun betrachten wir den widersprüchlichen Tracking-Fall. Für $\dot{x}_{ref} \neq 0$ und wenn nach wie vor (41) verwendet wird, wird (39) zu

$$\dot{\tilde{V}}_{real} = \dot{x}_{ref}^T \Psi E_T f_{des}, \quad (46)$$

wofür Gleichung (32) zu

$$\dot{V} = \dot{V}_{des} + \underbrace{\dot{x}_{ref}^T \Psi E_T f_{des}}_{\dot{\tilde{V}}_{cmd}} + \dot{\tilde{V}}_{real} \quad (47)$$

wird. Für inkonsistente Task-Referenzgeschwindigkeiten wird $\dot{\tilde{V}}_{cmd}$ typischerweise nicht zu Null, wodurch ein formaler Passivitätsbeweis für den Tracking-Fall erschwert wird (nicht Inhalt dieses Papers).

E. Behandlung von Unteraktuierung und anderen Aktuierungsbeschränkungen

[0094] Die im vorigen Kapitel präsentierten analytischen Lösungen sind für Regelungsprobleme bestimmt, welche von einem voll aktuierten Roboter ausgehen, dessen Aktuierungsbeschränkungen (oder andere Beschränkungen) nicht relevant sind. Dieses Kapitel wird die Fälle von Unteraktuierung und Aktuierungsbeschränkungen behandeln, und eine Lösung für solche Probleme im Kontext von MPTC vorschlagen.

[0095] 1) Unteraktuierung: Im Gegensatz zu Fixed-Base-Robotern ist die Roboter-Basis von sogenannten Free-Floating-Robotern (z.B. Humanoiden) nicht aktuiert. Um ein gewisses Maß an Kontrollierbarkeit zu erlangen, muss ein Free-Floating-Roboter seine End-Effektoren benutzen, um Kontaktkräfte zu erzeugen, welche das Fehlen einer direkten Basis-Aktuierung kompensieren. Die entsprechende Aktuierungs-Mapping-Matrix U aus Kapitel III-D hat die folgende Form:

$$U = \left[S^T, L_{\varepsilon\varepsilon}^T \right]. \quad (48)$$

[0096] Die entsprechenden Aktuierungs-Freiheitsgrade / Optimier-ungsvariablen sind

$$u_{cmd} = \begin{bmatrix} \tau_{j,cmd} \\ \omega_{EE,cmd} \end{bmatrix}, \quad (49)$$

wobei $\tau_{j,cmd}$ die kommandierten Gelenkdrehmomente bezeichnet und $w_{EE,cmd}$ die kommandierten End-Effektor-Kräfte sind. Die Gleichungen (48) und (49) basieren auf Elementen von Gleichung (2). Allerdings ist zu beachten, dass wir hier die kombinierten Link-Jabobians L_{all} und Link-Kräfte w_{all} durch eine entsprechende Auswahl ersetzt haben, die den End-Effektoren („EE“), welche sich im Kontakt befinden können, entspricht, also durch L_{EE} und $w_{EE,cmd.}$

[0097] 2) Kontakt-Zwangsbedingungen und Aktuierungsbeschränkungen: Die gerade eingeführten kommandierten End-Effektor-Kräfte $w_{EE,cmd}$ unterliegen oftmals ungleichungs-basierten Beschränkungen (den sogenannten Kontakt-Zwangsbedingungen). Beispielsweise werden diese Kontakt-Zwangsbedingungen in Anwendungen, welche sich auf robotisches Gehen beziehen (siehe Kapitel V-B), typischerweise in Form von Unilateralitäts- und Reibkegel-Zwangsbedingungen ausgedrückt. Die

Nichtberücksichtigung von solchen Kontakt-Zwangsbedingungen kann zu einem Versagen des Roboters führen. Aufgrund der physikalischen Beschränkungen des Roboters macht es oftmals Sinn, zusätzlich auch die kommandierten Gelenkdrehmomente $\tau_{j,cmd}$ zu beschränken. Auf diese Weise kann eine Aktuator-Saturierung vermieden werden.

[0098] 3) Lösung durch Quadratic Programming (QP): Eine beliebte Methode, welche die Behandlung der genannten Probleme bezüglich Unteraktuierung, Aktuator-Beschränkungen und Kontakt-Zwangsbedingungen ermöglicht, besteht darin, ein Quadratisches Programm (QP) in folgender Form (Expliziertere QP Formulierungen finden sich beispielsweise in [21, 15, 10].) aufzustellen

$$\underset{u_{cmd}}{\text{minimiere}}\ G_{QP} = \frac{1}{2} u_{cmd}^T T_u^T W T_u u_{cmd} - f_{des}^T W T_u u_{cmd},$$

unter Berücksichtigung von:

Kontakt – und Gelenkdrehmoments – Beschränkungen. (50)

[0099] Hierbei wurde die dritte Zeile von (34) für die Formulierung der QP Kostenfunktion G_{QP} entsprechend angepasst. Es ist wohlbekannt, dass ein QP dieselbe Lösung wie eine gewichtete pseudo-inversen-basierte Optimierung erzielt, solange die Beschränkungen nicht aktiv sind. Wir empfehlen, dass die exakt selbe passivitäts-garantierende Optimierungs-Gewichtung (PWOW) verwendet wird, welche in Kapitel III-D für die analytische Optimierung einsetzt wurde, also $W = \Lambda^{-1}\,\Psi$. Man beachte allerdings, dass die Passivität, sobald die Beschränkungen aktiv werden, nicht mehr garantiert werden kann; nicht einmal für den Regulations-Fall.

IV. EINORDNUNG IM BEZUG AUF VERWANDTE REGLERTYPEN UND GRANULARITÄT DER ANGESTREBTEN ENTKOPPLUNG

A. Vergleich von Inverse Dynamics (ID) und MPTC

[0100] In diesem Kapitel vergleichen wir die Zielsetzungen von Inverse Dynamics (ID) basierten Reglern und MPTC-basierten Reglern, und zeigen bedeutende Ähnlichkeiten auf.

Inverse Dynamics (ID):

[0101] Inverse Dynamics (ID) basierte Regler implementieren typischerweise eine stabile Dynamik zweiter Ordnung von folgender Form

$$\ddot{x}_k = \ddot{x}_{k,ref} + \hat{D}_{ID,k}\underbrace{\left(\dot{x}_{k,ref} - \dot{x}_k\right)}_{\dot{\tilde{x}}_k} + \hat{P}_{ID,k}\underbrace{\left(x_{k,ref} - x_k\right)}_{\tilde{x}_k}, \tag{51}$$

welche für positiv definite Proportional- und Ableitungs-Verstärkungs-Matrizen $\hat{P}_{ID,k}$ und $\hat{D}_{ID,k}$ das Tracking der Referenztrajektorie $x_{k,ref}$, $\dot{x}_{k,ref}$, $\ddot{x}_{k,ref}$,ermöglicht. Die beiden Verstärkungs-Matrizen werden typischerweise als Diagonalmatrizen gewählt, um eine vollständig entkoppelte, lineare Dynamik zu erhalten. Oftmals wird hierbei das sogenannte Pole Placement [1] verwendet, um ein kritisch gedämpftes Einschwingverhalten zu erzielen. Durch Einsetzen der Task-Space-Beschleunigung $\ddot{x}_k$ aus (5) in (51) und Auflösen nach dem Term, welcher sich auf die generalisierten Kräfte τ bezieht, ergibt sich:

$$J_k M^{-1}\tau = J_k M^{-1}\tau_g + Q_k\dot{q} + \ddot{x}_{k,ref} + \hat{D}_{ID,k}\dot{\tilde{x}} + \hat{P}_{ID,k}\tilde{x}_k. \tag{52}$$

[0102] Man beachte, dass wir hier nicht direkt nach τ auflösen, da abhängig vom gewählten Regler-Setup die Matrix J_k in manchen Fällen nicht invertierbar ist.

Modularer Passiver Tracking Controller (MPTC):

[0103] Durch Ersetzen von $f_{k,des}$ in (17) durch $f_k = T_k\tau$ (aus (16)), Vormultiplizieren des Ergebnisses mit M_k^{-1} und Auflösen finden wir die folgende gewünschte Task-Regler-Formulierung:

$$J_k M^{-1}\tau = J_k M^{-1}\tau_g + Q_k\dot{q} + \ddot{x}_{k,ref} + \underbrace{M_k^{-1}\left(C_k + D_k\right)}_{\hat{D}_{MPTC,k}}\dot{\tilde{x}} + \underbrace{M_k^{-1}K_k}_{\hat{P}_{MPTC,k}}\tilde{x}_k. \tag{53}$$

[0104] Beim Vergleich von (53) und (52) zeigt sich, dass die Grundstruktur des vorgestellten Modularen Passiven Tracking Controllers (MPTC) und die von Inverse Dynamics (ID) basierten Reglern die selbe ist. Sie unterscheiden sich lediglich durch die gewählten Proportional- und Ableitungs-Verstärkungs-Matrizen. Für ID basierte Regler werden $\hat{P}_{ID,k}$ und $\hat{D}_{ID,k}$ als konstant gewählt, während im Falle von MPTC $\hat{P}_{MPTC,k}$ und $D_{MPTC,k}$ jeweils eine Funktion der aktuellen Task-Trägheit M_k sind. Zusätzlich enthält $\hat{D}_{MPTC,k}$ die Task-Coriolis-Matrix C_k, sodass die taskbezogenen Coriolis-Terme nicht gekürzt werden, was eine Voraussetzung für Passivität ist und die Robustheit erhöht (aufgrund der verringerten Anzahl an Feedback-Kanälen, welche Probleme im Bezug auf Sensorrauschen und Modellfehler bewirken können.).

[0105] Ein wichtiger Aspekt hierbei ist, dass (52) und (53) hier im Task-Beschleunigungs-Space geschrieben sind, in welchem sich für ID konstante Verstärkungsmatrizen ergeben, welche konstanten Systembzw. Task-Eigenwerten entsprechen, während die lokalen Eigenwerte von MPTC konfigurationsabhängig sind. Wenn hingegen (52) und (53) durch Vormultiplizieren mit M_k in den Task-Kraft-Space transferiert

werden, zeigt sich, dass die gefühlte Dämpfung und Steifigkeit im Falle von MPTC konstant sind, während sie für ID konfigurationsabhängig sind.

[0106] Anmerkung: Die starke Ähnlichkeit von (52) und (53) kann sich vorteilhaft bezüglich einer möglichen Portierung von bestehenden ID-basierten Reglerframeworks in die MPTC Methodik auswirken.

B. Vergleich mit PD+ Regelung / passivitäts-basierter Ganzkörperregelung

[0107] Durch Neuanordnung der Terme in der Subtask-Regler-Zielsetzung (17), welche sich auf Coriolis- und Zentrifugal-Effekte (ab hier durch „CC“ abgekürzt) beziehen, finden wir die folgende Gleichung:

$$C_k \dot{\tilde{x}}_k + M_k Q_k \dot{q} = C_k \dot{x}_{k,ref} + \underbrace{M_k Q_k \left(\underbrace{I - T_k^T j_k}_{N_k} \right)}_{B_k}. \tag{54}$$

[0108] Hier bezeichnet N_k einen Nullraum-Projektor, welcher alle Komponenten von $\dot{q}$ entfernt, welche einen Effekt auf die Task-Space-Geschwindigkeit $\dot{x}_k$ haben würden, und die Matrix B_k kombiniert die entsprechenden CC-bezogenen Feedback-Terme. Durch Einsetzen von (54) in (17) finden wir eine alternative (jedoch gleichwertige) Subtask-Reglerformulierung:

$$f_{k,des} = T_k \tau_g + B_k \dot{q} + M_k \ddot{x}_{k,ref} + C_k \dot{x}_{k,ref} + D_k \dot{\tilde{x}}_k + K_k \tilde{x}_k. \tag{55}$$

[0109] Im Vergleich zur ursprünglichen Subtask-Reglerformulierung aus (17) ermöglicht diese Formulierung einen direkteren Einblick in die CC-Term-Kürzungsstrategie des MPTC Reglers: MPTC kürzt alle nicht-task-bezogenen CC-Feedbackterme (vergleichbar mit den nicht-diagonalen Elementen in [7]) und generiert einen task-bezogenen Feedforward-Term $C_k \dot{x}_{k,ref}$, während die taskbezogenen CC-Feedbackterme nicht gekürzt bzw. benutzt werden. Letzteres kann durch Betrachtung des nominellen Subtask-Closed-Loop-Verhaltens (20), in welchem die taskbezogenen CC-Terme unberührt bleiben, noch einmal verifiziert werden. Das Behalten der taskbezogenen CC-Terme ist eine Voraussetzung für Passivität und führt zu einer höheren Robustheit gegenüber Modellfehlern im Vergleich zu Inverse Dynamics basierten Reglern. Letztere kürzen alle CC-Effekte komplett (basierend auf dem potenziell ungenauen Robotermodell), während MPTC-basierte Regler nur die nicht-task-bezogenen CC-Terme kürzen.

[0110] Ein interessanter Spezialfall liegt vor, wenn nur eine einzige Task betrachtet wird (welche alle Subtasks durch Stapeln der entsprechenden Subtask-Jacobians in eine Gesamt-Task kombiniert) und die Task-Jacobian J_k quadratisch und invertierbar ist. Voraussetzung hierfür ist, dass sich die Subtasks nicht widersprechen. In diesem speziellen Fall ist $B_k = 0$ und deshalb ist (55) äquivalent zu einem klassischen PD+ Regler [19] (siehe **Abb.** 1) oder (im Falle von Ungleichungs-Beschränkungen) zu den später eingeführten passivitätsbasierten Ganzkörperreglern (WBC) [12, 16].

C. Granularität der angestrebten Entkopplung

[0111] MPTC stellt ein generisches Reglerdesignwerkzeug dar, sowohl bezüglich der Wahl der Sub-Regler-Typen (z.B. kartesische oder Gelenk-Regler) als auch bezüglich der „Granularität“ der angestrebten Entkopplung (Man beachte: wir sprechen hier über „angestrebte Entkopplung“. Im Falle von konsistenten Tasks bewirkt MPTC eine strikt entkoppelte Fehlerdynamik. Im Falle von Inkonsistenzen hingegen ist eine Verkopplung der entsprechenden Task-Fehlerdynamiken unvermeidbar (zumindest für weiche Taskpriorisierung).). Wenn wir in diesem Paper von Subtasks sprachen, so hatten wir bisher nie deren Dimensionalität, d.h. die Anzahl an Freiheitsgraden, welche der jeweiligen Task zugeordnet sind, weiter spezifiziert. Das bedeutet, es existiert eine Vielzahl an vorstellbaren Reglerkonfigurationen. Wenn man verschiedene Regel-Ziele (z.B. ein kartesisches End-Effektor-Tracking-Ziel und ein Gelenkregelungs-Ziel) in einem einzigen Stack of Tasks (SoT, Aufgabenstapel) kombiniert, so bleibt vorerst die Frage offen, welcher Teil des SoT welcher MPTC-bezogenen Subtask zugeordnet wird. Als ein Beispiel: eine sechsdimensionale kartesische Task kann als zwei entkoppelte (linear und angular) Regelungs-Subtasks definiert werden, oder alternativ als eine einzige sechsdimensionale Task. Im ersten Fall wären die lineare und angulare Fehlerdynamik voneinander entkoppelt, während sie im zweiten Fall verkoppelt wären.

[0112] Man kann sich außerdem dazu entscheiden, alle Reglerzielsetzungen in eine einzige Task zu kombinieren (was in einem Voll Verkoppelten Passiven Tracking-Regler (FC-PTC) resultiert), was unter bestimmten Umständen äquivalent zu einem PD+ Regler ist (siehe Kapitel IV-B und **Abb.** 1). Alternativ kann man ein Set von multiplen Tasks (z.B. eine Task für den linken Fuß, eine für den rechten) definieren, welches durch das Standard-MPTC-Framework behandelt werden kann. Schließlich kann man sich auch für ein maximal granulares Regler-Setup entscheiden, in welchem jede Zeile des SoT eine eigene Task darstellt. Diesen speziellen Typ des Regler-Setups bezeichnen wir als Voll Entkoppelten Passiven Tracking Controller (FD-PTC). Von allen möglichen MPTC Setups ist FD-PTC dasjenige, welches einem Inverse Dynamics Regler am meisten ähnelt;

Beide basieren auf ein-Freiheitsgrad entkoppelten Task-Dynamiken (siehe **Abb.** 1). Allerdings ist anzunehmen, dass FD-PTC eine höhere Robustheit aufweist, da hierbei die Trägheit nicht modifiziert wird und weniger CC-Terme gekürzt werden.

[0113] Wie bereits oben erwähnt weist jede Task nur einen einzigen Gewichtungs-Skalar ψ_k auf, welcher es erlaubt, die Tasks (weich) gegeneinander zu priorisieren. Dies reduziert den nötigen Tuning-Aufwand. Es ist allerdings zu beachten, dass, um die Priorität einer bestimmten Task-Komponente zu erhöhen (z.B. wenn die z-Richtung den x und y Komponenten einer kartesischen Task vorgezogen werden soll), dieser eine eigene Task und eine geeignete Gewichtung (z.B. ψ_z) zugewiesen werden muss.

V. SIMULATIONEN UND EXPERIMENTE

[0114] Um die Leistungsfähigkeit unseres vorgeschlagenen MPTC-Frameworks zu validieren, haben wir mehrere Simulationen durchgeführt: einerseits einfache Simulationen basierend auf einer Vorwärtsintegration von (3), die die berechneten Reglerdrehmomente als Eingabe verwendeten, andererseits Simulationen des humanoiden Roboters TORO [8] unter Verwendung von OpenHRP [14] als Simulationsumgebung.

A. Fixed-base Roboter-Simulationen

[0115] Die erste vorgestellte Simulation soll den Regulations-Fall für ein voll aktuiertes, vollständig bestimmtes und damit konfliktfreies Task-Setup untersuchen. Zu diesem Zweck wurden die sechs Gelenke eines fixed-base-Roboterarms (dessen Kinematik und Trägheitseigenschaften einem der Beine von TORO entsprechen) zwei verschiedenen Gelenkaufgaben A und B zugeordnet (welche jeweils drei Gelenke abdecken, siehe **Abb.** 2). Die entsprechenden Gelenk-Task-Steifigkeits- und -Dämfungs-Gains (Hinweis: Der Kürze halber werden hier die Einheiten für die Steifigkeit (linear: $\frac{N}{m}$, angulär: $\frac{Nm}{rad}$, und Dämpfung (linear: $\frac{Ns}{m}$, angulär: $\frac{Nms}{rad}$) weggelassen.) wurden als $K_A = 300I_{3\times3}$, $D_A = \text{diag}([40, 20, 10])$, $K_B = 40I_{3\times3}$ und $D_B = 2I_{3\times3}$ gewählt. Man beachte, dass die skalaren Aufgabengewichte ψ_k (hier alle auf 1 gesetzt) keine Auswirkung haben, da aufgrund fehlender Konflikte alle Tasks ohnehin perfekt erfüllt wurden. Es ist zu beachten, dass der Controller kein zusätzliches Tuning benötigte. Die dargestellten Parameter wurden ausgewählt, um den didaktischen Wert von **Abb.** 2 zu verbessern. Zu Beginn der Simulation konvergieren alle Gelenke schnell zu den anfänglichen Sollwerten. Nach einer Sekunde waren die Gelenke der Gruppe A einer Geschwindigkeitsänderung ausgesetzt, was zu einem Sprung in den entsprechenden Task-Fehler-Ableitungen führte. Man beachte, dass Gruppe B, während Gruppe A zurückkonvergiert, aufgrund der Task-Entkopplung völlig unberührt bleibt. Nach zwei Sekunden werden die Sollwerte der Gruppe B geändert. Auch hier konvergieren die entsprechenden Gelenkkoordinaten, während Gruppe A nun völlig unberührt bleibt. Hinweis: Die hier beobachtete perfekte Task-Entkopplung war nur aufgrund der Konsistenz der beiden Gelenk-Tasks erreichbar. Schließlich wird nach drei Sekunden ein Drehmoment-Offset von $\tau_{int} = [20, 20, 20, 5, 5, 5]$Nm zuerst angelegt und dann nach Sekunde 4 entfernt. Auch hier ist das beobachtete Controller-Verhalten gutmütig und erfüllt unsere Erwartungen. Mit dieser Simulation können wir verifizieren, dass der Gesamtwert der Lyapunov-Funktion V immer ab-nimmt, außer im Falle von Störungen und Sollwertänderungen (siehe **Abb.** 2 (rechts)).

[0116] Die zweite vorgestellte Simulation evaluiert die Leistungsfähigkeit des Reglers für ein widersprüchliches Task-Setup, wiederum für den Regulations-Fall. Diesmal wurde eine kartesische Sechs-Freiheitsgrade-Endeffektor-Task mit einer Steifigkeit von $K_{cart} = \text{diag}([2000, 2000, 2000, 100, 100, 100])$ und einer Dämpfung von $D_{cart} = \text{diag}([500, 500, 500, 20, 20, 20])$ zu den zuvor beschriebenen Gelenk-Tasks A und B hinzugefügt. Die skalaren Taskgewichte ψ_k werden auf 1 gesetzt. Nach dem Starten des Reglers konvergiert der Roboter in eine Gleichgewichtsposition. Man beachte, dass der Gesamtwert der Lyapunov-Funktion V monoton abnimmt, während die gewichteten Lyapunov-Funktionswerte der Sub-Tasks auch zunehmen können (wie z.B. $\psi_{j_a} \cdot V_{j_a}$ in **Abb.** 3).

B. Humanoide Robotersimulationen und Experimente

[0117] Schließlich führten wir Gang-Simulationen des humanoiden Roboters TORO durch (siehe **Abb.** 4), die auf einem hybriden WBC-Setup basierten: Inverse Dynamics-basierte Torso-Orientierungs- und Ganzkörper-Posen-Tasks, eine Task für die Regelung der Divergent Component of Motion (DCM) [9] und Regularisierung des Drehimpulses, sowie kartesische (6 Freiheitsgrade) MPTC-basierte Regler für das Fuß-Tracking wurden gemischt (Hinweis: Aufgrund der Modularität von MPTC können ID-basierte und MPTC-basierte Aufgaben im selben Ganzkörperregelungs-Setup kombiniert werden.), Die Fußreferenztrajektorien werden präzise getrackt. Nach 2.5s wird der linke Fuß durch eine Kraft von -60N in x-Richtung für 0.3s gestört, was zu einem maximalen Fußpositionsfehler von 51mm und einem maximalen DCM-Fehler von 45mm führt . Nachdem die Störung beseitigt ist, konvergiert der Fuß schnell genug, um erfolgreich weiterzugehen.

[0118] Um die Leisungsfähigkeit von MPTC unter realen Bedingungen zu bewerten, haben wir mehrere Experimente mit TORO durchgeführt, darunter Push Recovery (stehend), Mensch-Roboter-Interaktion und Gehen (ein entsprechendes Video findet sich hier: https://youtu,be/WdF9UQK8alo), Hier präsentieren wir ein Gangexperiment, bei dem TORO sechs Schritte vorwärts läuft (siehe **Abb.** 5 für Details), Separate MPTC-basierte Controller wurden zur Regelung der Füße (kartesisches Sechs-Freiheitsgrade-Tracking), des Rumpfes (Drei-Freiheitsgrade-Rotations-Tracking), der Beingelenke, des Taillengelenks und der Oberkörpergelenke verwendet (die letzten drei dienten zur Regelung der Gesamtkörperpose und als Regularisierungs-Tasks). Genau wie für die Simulationen wurden zusätzlich ein DCM-basierter Regler und ein Drehimpuls-Regularisierungsregler angewendet. Die Gang-Performance war insgesamt robust. Es können jedoch Trackingfehler beobachtet werden, die unserer Ansicht nach durch Drehmoment-Offsets und Gelenkreibung verursacht werden. Im Vergleich zu ID-basierten Reglern [10] war der Tuning-Aufwand gering, was die Robustheit von MPTC beim Einsatz unter realen Bedingungen belegt.

VI. ZUSAMMENFASSUNG UND AUSBLICK

[0119] In dieser Arbeit wurde der sogenannte Modulare Passive Tracking Controller (MPTC) eingeführt. Dieser generische Regler beabsichtigt in einem ersten Schritt, mehrere Subtask-Zielsetzungen unabhängig voneinander zu erfüllen. Diese primär unabhängigen Subtask-Regler werden dann zu einem Gesamtregler zusammengeführt. Das Design und die Analyse des Reglers basieren auf der Lyapunov-Theorie, welche Aussagen über Stabilität und Passivität erleichtert. Einer der Hauptbeiträge dieses Papers ist der Entwurf einer Optimierungsgewichtungsmatrix, die für voll-aktuierte Roboter die Passivität eines kompletten Sets von widersprüchlichen Subtasks garantiert. Das vorgeschlagene Regelungs-Framework wurde in mehreren Simulationen und Experimenten für fixed-base und free-floating Roboter validiert.

[0120] In unserer zukünftigen Forschung wollen wir die Regelungskonzepte von Inverse Dynamics (ID), des vorgeschlagenen Modularen Passiven Tracking Controllers (MPTC) und von anderen passivitätsbasierten Reglern, wie etwa dem PD+ Regler, ausführlich vergleichen. Dieser umfassende Vergleich wird auf theoretischen Analysen, Simulationen und Hardwareexperimenten basieren.

REFERENCES

[1] J. Ackermann. Parameter space design of robust control systems. IEEE Transactions on Automatic Control, 25(6):1058-1072, December 1980. ISSN 2334-3303. doi: 10.1109/TAC.1980.1102505.

[2] G. Antonelli. Stability Analysis for Prioritized closed-loop inverse kinematic algorithms for redundant robotic systems. IEEE Transactions on Robotics, 25(5):985-994, 2009.

[3] K. Bouyarmane and A. Kheddar. On Weight-Prioritized multitask control of humanoid robots. IEEE Transactions on Automatic Control, 63 (6):1632-1647, June 2018. ISSN 2334-3303. doi: 10.1109/TAC.2017.2752085.

[4] S. Chiaverini. Singularity-robust task-priority redundancy resolution for real-time kinematic control of robot manipulators. IEEE Transactions on Robotics and Automation, 13 (3):398-410, June 1997. ISSN 2374-958X. doi: 10.1109/70.585902.

[5] A. Dietrich. Whole-Body Impedance Control of Wheeled Humanoid Robots, volume 116. Springer International Publishing, 2016. ISBN 978-3-319-40557-5.

[6] A. Dietrich and C. Ott. Hierarchical Impedance-Based tracking control of kinematically redundant robots. IEEE Transactions on Robotics, 36(1):204-221, 2020. ISSN 1941-0468. doi: 10.1109/TRO.2019.2945876.

[7] A. Dietrich, C. Ott, and A. Albu-Schäffer. Multi-objective compliance control of redundant manipulators: hierarchy, control, and stability. In IEEEIRSJ Int. Conf. on Intelligent Robots and Systems, pages 3043-3050, Nov 2013. doi: 10.1109/IROS.2013.6696787.

[8] J. Englsberger, A. Werner, C. Ott, B. Henze, M. A. Roa, G. Garofalo, R. Burger, A. Beyer, O. Eiberger, K. Schmid, and A. Albu-Schäffer. Overview of the torque-controlled humanoid robot TORO. In IEEE-RAS Int. Conf. on Humanoid Robots, pages 916-923, Nov 2014. doi: 10.1109/HUMANOIDS.2014.7041473.

[9] J. Englsberger, C. Ott, and A. Albu-Schäffer. Three-Dimensional Bipedal Walking Control Based on divergent component of motion. IEEE Transactions on Robotics, 31(2):355-368, April 2015. ISSN 1941-0468. doi: 10.1109/TRO.2015. 2405592.

[10] J. Englsberger, G. Mesesan, A. Werner, and C. Ott. Torque-Based Dynamic Walking - a long way from simulation to experiment. In IEEE Int. Conf. on Robotics and Automation (ICRA), pages 440-447, 2018.

[11] Adrien Escande, Nicolas Mansard, and Pierre-Brice Wieber. Hierarchical quadratic programming: fast online humanoid-robot motion generation. The International Journal of Robo-

tics Research, 33(7):1006-1028, 2014. doi: 10.1177/ 0278364914521306.

[12] B. Henze, M. A. Roa, and Ch. Ott. Passivity-based whole-body balancing for torque-controlled humanoid robots in multi-contact scenarios. The International Journal of Robotics Research, 35(12):1522-1543, 2016. doi: 10.1177/ 0278364916653815.

[13] M. A. Hopkins, D. W. Hong, and A. Leonessa. Compliant locomotion using whole-body control and divergent component of motion tracking. In IEEE Int. Conf. on Robotics and Automation (ICRA), pages 5726-5733, May 2015. doi: 10.1109/ ICRA.2015.7140001.

[14] F. Kanehiro, H. Hirukawa, and S. Kajita. Openhrp: Open Architecture Humanoid Robotics Platform. The International Journal of Robotics Research, 23(2):155-165, 2004. doi: 10.1177/0278364904041324.

[15] T . Koolen, S. Bertrand, G. Thomas, T. de Boer, T. Wu, J. Smith, J. Englsberger, and J. Pratt. Design of a Momentum-Based Control Framework and application to the humanoid robot atlas. International Journal of Humanoid Robotics, 13(01):1650007, 2016. doi: 10.1142/S0219843616500079.

[16] G. Mesesan, J. Englsberger, G. Garofalo, C. Ott, and A. Albu-Schäffer. Dynamic walking on compliant and uneven terrain using dcm and passivity-based whole-body control. In IEEE-RAS 19th Int. Conf. on Humanoid Robots (Humanoids), pages 25-32, 2019.

[17] R. M. Murray. A mathematical introduction to robotic manipulation. CRC press, 2017.

[18] Y. Nakamura, H. Hanafusa, and T. Yoshikawa. Task- priority based redundancy control of robot manipulators. The International Journal of Robotics Research, 6(2):3-15, 1987. doi: 10.1177/027836498700600201.

[19] B. Paden and B. Riedle. A Positive-Real Modification of a class of nonlinear controllers for robot manipulators. In 1988 American Control Conference, pages 1782-1785, June 1988. doi: 10.23919/ACC.1988.4790015.

[20] J. Peters, M. Mistry, F. Udwadia, J. Nakanishi, and S. Schaal. A unifying framework for robot control with redundant dofs. Autonomous Robots, 24:1-12, 2008. doi: 10.1007/s10514-007-9051-x.

[21] L. Righetti, J. Buchli, M. Mistry, and S. Schaal. Inverse dynamics control of floating-base robots with external constraints: A unified view. In IEEE Int. Conf. on Robotics and Automation, pages 1085-1090, May 2011. doi: 10.1109/ICRA.2011. 5980156.

[22] L. Sentis and O. Khatib. Synthesis of Whole-Body behaviors through hierarchical control of behavioral primitives. International Journal of Humanoid Robotics, 2(4):505-518, 2005.

[23] L. Sentis, J. Park, and O. Khatib. Compliant Control of multicontact and center-of-mass behaviors in humanoid robots. IEEE Transactions on Robotics, 26(3):483-501, 2010.

[0121] Fig. 6 zeigt exemplarisch eine mögliche Anwendung des erfindungsgemäßen Verfahrens auf. Hierbei werden mehrere impedanzbasierte Reglermodule 102 (Modulare Passive Tracking Controller) durch das erfindungsgemäße Verfahren in einen Gesamtregler 104 integriert. Im gezeigten Beispiel umfasst die Darstellung der Reglermodule 102 jeweils eine Referenztrajektorie und eine Masse, die über eine Feder-Dämpfer-System miteinander verbunden sind, wobei die Reglermodule 102 mit den Bezeichnungen A und C jeweils ein kartesisches Regelungs-Task darstellen, während die Reglermodule 102 mit den Bezeichnungen B und D als Gelenk-Regler formuliert sind. Wie in **Fig.** 6 gezeigt können die verschiedenen Reglermodule 102, welche jeweils aus einer Task-Mapping-Matrix und einem Task-Vektor bestehen, zu einem Stack of Tasks bestehend aus einer vereinigten Task-Mapping-Matrix 106 sowie aus einem kombinierten Task-Vektor 110 zusammengefasst werden. Durch ein geeignetes Optimierungsverfahren, beispielsweise mithilfe von Optimierungsvariablen 108, können die durch den Stack of Tasks repräsentierten Regler-Module zu einem Gesamtregler 104 fusioniert werden. Dabei wird ein optimaler Kompromiss aus den verschiedenen Unter-Regler-Zielsetzungen erzielt, wobei insbesondere die Passivität des resultierenden Gesamtreglers 104 bestmöglich umgesetzt wird.

Bezugszeichen

100	Roboter
102	Reglermodul
104	Gesamtregler
106	Task-Mapping-Matrix
108	Optimierungsvariablen
110	Task-Vektor

Patentansprüche

1. Verfahren zum Regeln eines kinematisch redundanten Roboters (100), um mehrere Tasks zu erfüllen, wobei wenigstens ein passivitäts-basiertes erstes Reglermodul (102) verwendet wird, für das wenigstens eine erste Reglermodul (102) wenigstens eine Task-Zielbeschreibung und wenigstens

ein zugehöriges Task-Mapping berechnet werden, für die Tasks wenigstens eine Gewichtung in Form einer task-spezifischen symmetrischen Matrix $W = \Psi_k M_k^{-1}$ berechnet wird, wobei sich die Matrix W_k aus der Inversen der Task-Trägheitsmatrix M_k und einem Gewichtungs-Skalar Ψ_k zur unabhängigen Gewichtung verschiedener Tasks untereinander zusammensetzt, und das wenigstens eine erste Reglermodul (102) unter Verwendung der wenigstens einen Task-Gewichtungs-Matrix W_k, welche in Form von Sub-Matrizen in eine symmetrische und block-diagonale Gesamt-Gewichtungs-Matrix W zusammengeführt wird, in einen Gesamtregler (104) integriert wird.

2. Verfahren nach Anspruch 1, **dadurch gekennzeichnet**, dass die wenigstens eine Task-Zielbeschreibung und das wenigstens eine zugehörige Task-Mapping derart berechnet werden, dass ein nominelles Verhalten des wenigstens einen ersten Reglermoduls (102) dem Verhalten eines Feder-Masse-Dämpfer-Systems entspricht.

3. Verfahren nach wenigstens einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass die wenigstens eine Task-Zielbeschreibung und das wenigstens eine zugehörige Task-Mapping basierend auf einer natürlichen Roboter-Trägheit berechnet werden.

4. Verfahren nach wenigstens einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass die wenigstens eine Task-Zielbeschreibung und das wenigstens eine zugehörige Task-Mapping derart berechnet werden, dass taskspezifische Coriolis- und Zentrifugal-Effekte ungeändert bleiben, während alle übrigen Coriolis- und Zentrifugal-Effekte kompensiert werden.

5. Verfahren nach wenigstens einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass die wenigstens eine Task-Zielbeschreibung als Task-Vektor und/oder das wenigstens eine zugehörige Task-Mapping als Task-Mapping-Matrix berechnet werden/wird.

6. Verfahren nach wenigstens einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass wenigstens ein weiteres Reglermodul in den Gesamtregler (104) integriert wird.

7. Verfahren nach wenigstens einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass wenigstens ein als Zwangsbedingung formuliertes weiteres Reglermodul in den Gesamtregler (104) integriert wird.

8. Verfahren nach wenigstens einem der vorhergehenden Ansprüche 6 bis 7, **dadurch gekennzeichnet**, dass das wenigstens eine weitere Reglermodul gewichtet wird.

9. Verfahren nach wenigstens einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass das wenigstens eine erste Reglermodul (102) und/oder das wenigstens eine weitere Reglermodul derart gewichtet werden, dass der Gesamtregler (104) auch bei einem überbestimmten Regelungsproblem ein zumindest annähernd passives Verhalten aufweist.

10. Verfahren nach wenigstens einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass das wenigstens eine erste Reglermodul (102) und/oder das wenigstens eine weitere Reglermodul mithilfe wenigstens einer Pseudo-Inversen und/oder wenigstens einer Inversen optimiert werden/wird.

11. Verfahren nach wenigstens einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass das wenigstens eine erste Reglermodul (102) und/oder das wenigstens eine weitere Reglermodul quadratisch optimiert werden/wird.

12. Verfahren nach wenigstens einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass wenigstens eines der Reglermodule (102) als Tracking-Regler ausgeführt wird.

13. Verfahren nach wenigstens einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass wenigstens eines der Reglermodule (102) als Regulation-Regler ausgeführt wird.

14. Verfahren nach wenigstens einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, dass jedem regelbaren Freiheitsgrad ein gesondertes Reglermodul (102) zugeordnet wird.

15. Computerprogrammprodukt, umfassend Befehle, die bei der Ausführung des Programms mithilfe wenigstens eines Prozessors diesen veranlassen, das Verfahren nach wenigstens einem der Ansprüche 1 bis 14 auszuführen.

Es folgen 5 Seiten Zeichnungen

Anhängende Zeichnungen

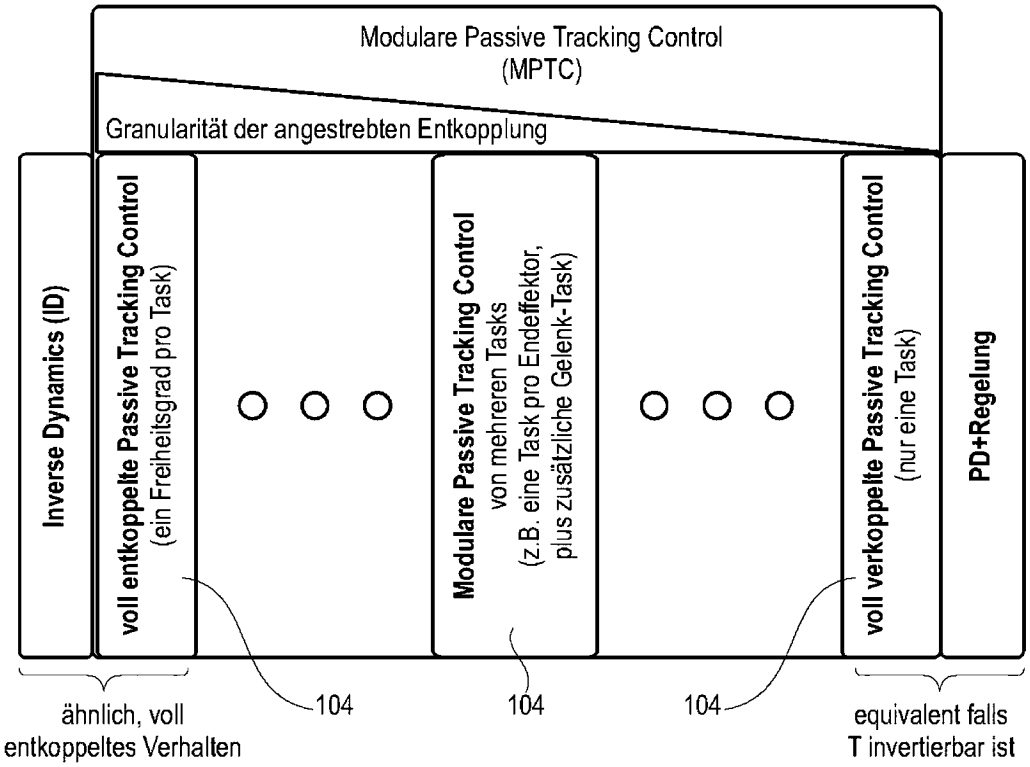


Fig. 1

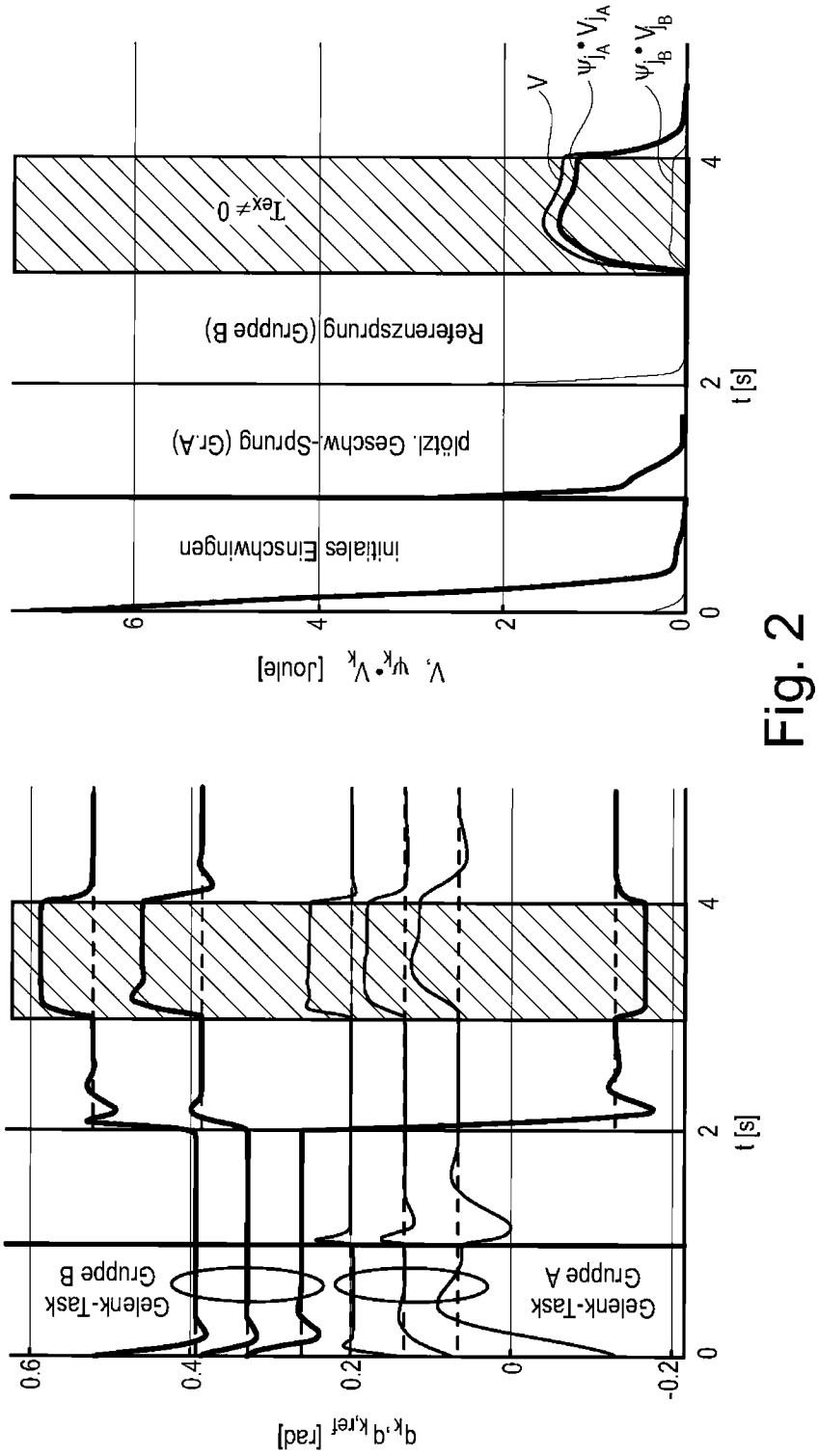

V, ψk • Vk [Joule]
initiales Einschwingen
plötzl. Geschw.-Sprung (Gr.A)
Referenzsprung (Gruppe B)
Tex ≠ 0
V
ψjA • VjA
ψjB • VjB
t [s]
qk ,qk,ref [rad]
Gelenk-Task Gruppe A
Gelenk-Task Gruppe B
t [s]

Fig. 2

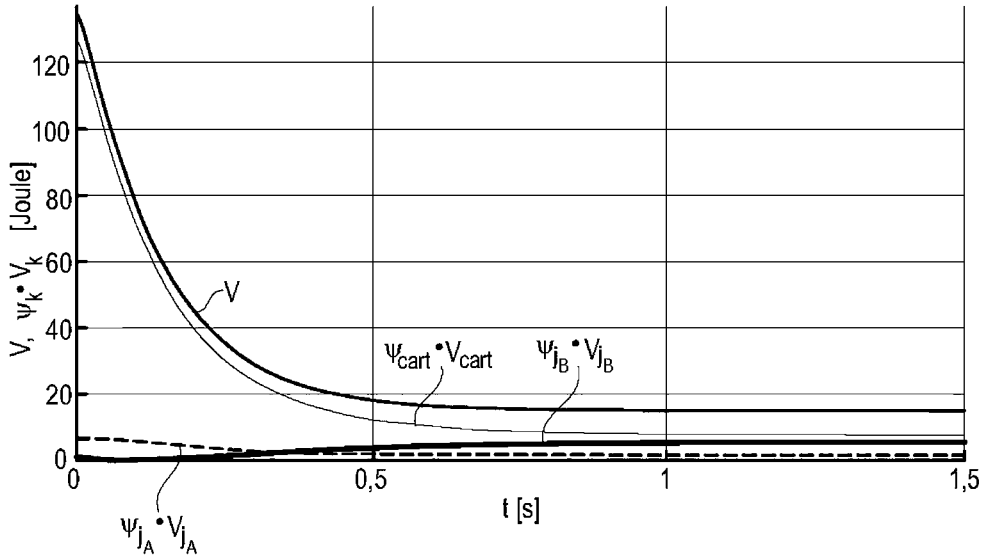

120
100
80
60
40
20
0
V, ψk • Vk [Joule]
V
ψcart • Vcart
ψjB • VjB
ψjA • VjA
0
0,5
1
1,5
t [s]

Fig. 3

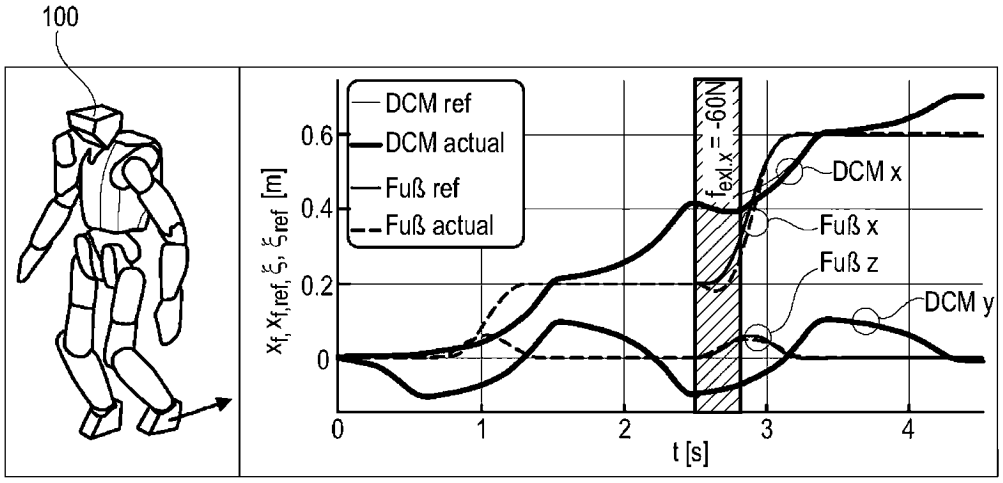

100
DCM ref
DCM actual
Fuß ref
Fuß actual
$x_f, x_{f,ref}, \xi, \xi_{ref}$ [m]
0.6
0.4
0.2
0
$f_{ext,x}$ = -60N
DCM x
Fuß x
Fuß z
DCM y
0
1
2
3
4
t [s]

Fig. 4

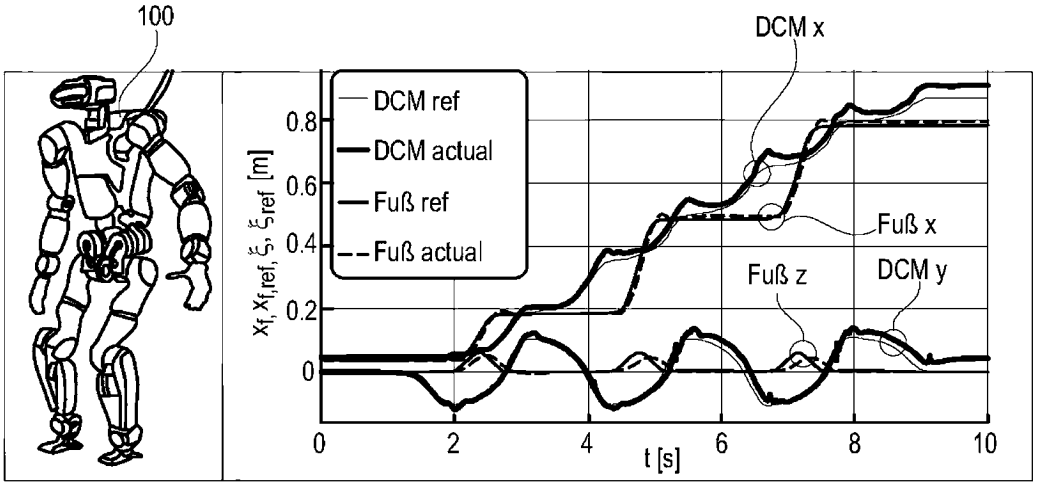

100
DCM x
DCM ref
DCM actual
Fuß ref
Fuß actual
$x_f, x_{f,ref}, \xi, \xi_{ref}$ [m]
0.8
0.6
0.4
0.2
0
Fuß x
Fuß z
DCM y
0
2
4
6
8
10
t [s]

Fig. 5

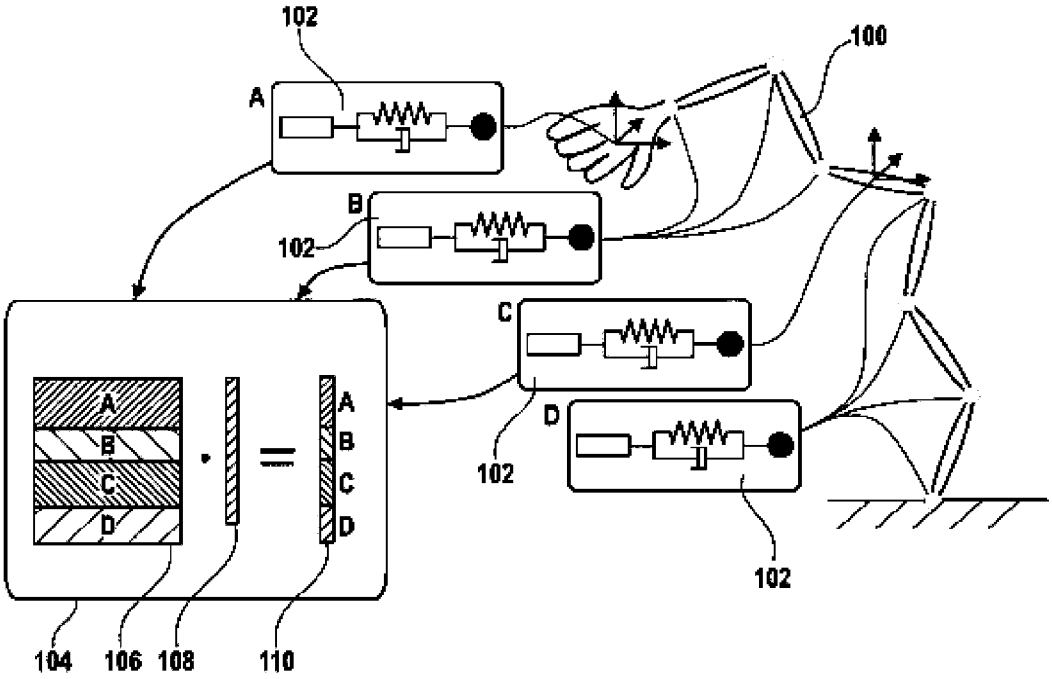


Fig. 6

 Das Dokument wurde durch die Firma Luminess hergestellt.