

Eigenschaften von ES

- in (große) Systeme integriert, dedizierte Fkt im Gesamtsyst.
- vor Anwender verst.
- auf Anwendung zugesch.
- informationsverarbeitend
- Sensoren/Aktoren zur Umwelt
- Automaten, Werkzeugmaschinen, Robotik, Kraftwerksteuer., Gebäudeautomat, Automob, Telekomm, Mikrowelle, Kamera (Consumer Elektronik)
- Implementierung: vollständig HW (Geschwindigkeit) bis vollständig SW auf Standardprozessor (flexibler/schnellerer Entwurf)
- Reaktive Systeme: ES als Regelung/Steuerung
- Transformierende Systeme: digitale Signalverarbeitung, einfache Berechnungen auf Datenstrom

Bsp Kfz:

- stark verteilte Steuerung, Kommunikation: BUS
- Echtzeitfähig, Fehlertoleranz
- Hardware von 8-bit mini-pc bis mehrere 64-bit-Prozessoren

Bsp TV:

- Signalverarbeitung (digital)

Moore's Law: 4x soviel Speicher alle 3 Jahre; Leistungsverdopplung alle 18 Monate

Anforderungen (consumer)

- so gut wie fehlerfrei (keine Updates (naja - Firmware Updates gibts schon recht häufig)/Wartung)
- intuitiv
- hw/sw immer im Bundle

Entwicklungsphasen:

- 1) HW in SW+Microproz. umwandeln
- 2) Hinzufügen neuer Eigenschaften zu bestehender SW (unsauber)
- 3) Neuentwurf (SW wurde zu groß, komplex)
- 4) Systemaspekte, Vernetzung, Produktgrenzen verschwimmen

Struktur eines ES (folie 01_4)

Umwelt - Techn. Sys mit Aktoren/Sensoren - Steuerung/vorverarbeitung - Weltmodell ... -

Mensch-Maschine-Schnittstelle

Robotik

Definitionen Roboter

Gelenke (Rotation: Drehachse senkrecht zu beiden Gliedern, Torsion: Drehachse parallel zu beiden, Revolver: senkrecht zum einem und parallel zum anderen Glied, Lineargelenk (Schubgelenk))

Arbeitsraum: kann tatsächl. angefahren werden.

Grundform: wenn man Gelenkwinkelgrenzen/gegenseitige Behinderung ignoriert

Koordinatensyst. kartesisch (LLL), Zylinder (LVL/LTL), Kugel (TRL)

Umrechnung: kart->Zylinder: r mit Pythagoras, alpha mit Tangens, ->Kugel: $\beta = z/r$,

Grundform Handgelenk: TRR (Mensch) oder TRT

Antriebe

Elektrisch (Platz, Genauigkeit/Kraft)

Pneumatisch (ungünstige Umgebungen, einfach/laut, Steuerung)

Hydraulisch (Kraft/Dreck, Platz, Genauigkeit, Reaktionszeit)

(Vorteile/Nachteile)

Koordinatensystem

Basis- Orts- rechts/linksdrehend

Ort, bezogen auf BKS: Ortsvektor

Orientierung, bezogen auf BKS: Rotationsmatrix

Zusammen LAGE

Transformationen von Koordinatensystemen (rotation, translation), Umrechnung zw. Koord. Systemen:

$p = p_{xyz} = ex * p_x \dots$

$p = puvw = eu * pu...$
also $px = p * ex = puvw * ex...$

Rotationsmatrixen, Drehung um x,y,z

(C -S)
(S C)
()

Eigenschaften von Rotationsmatrixen

Rot-Matr. gdw orthogonal (regulär $R^{-1} = R^T$), $\det(R) = 1$.

Euler-Winkel: Rotation um z, x', z'', $R_{ges} = R_z * R_x * R_z$ jede Rot. so beschreibbar, Herleitung

Roll-Pitch-Yaw: Angabe der Drehungen unabhängig voneinander bzgl. BKS $R_{ges} = R_z * R_y * R_x$

6-dim. Beschreibungsvektor (3 Ort, 3 Winkel)

homogene Koordinaten,

(R u) u=Ort, z=Zerrung=(000)

(z s) s=1, R= Rotation

A homogene Matrix =

(R u), R Rotation

(0 1)

Dann $A^{-1} = R^T * T^{-1}$ (also $T = (0u)$, $U^{-1} = (0 - u)$)

Ermittlung des Beschreibungsvektors aus einer homogenen Matrix

Translation trivial,

Rotation: $R = R_z * R'_x * R''_z \Leftrightarrow R_z^{-1} * R = R'_x * R''_z$

Dann Gleichungen bilden.

In der Regel: $\sin \alpha / \cos \alpha = a / b$ Da \arcsin / \arccos nicht eindeutig auf $-\pi$ bis π : $\text{atan2}(a,b) = \arccos(b)$:

falls $a > 0$; $-\arccos(b)$ sonst

Verkettete Lagebeschreibung.

H_C Lage von C bzgl. Koord. B; H_B Lage von B bzgl. Koord. A

Dann H_{Ges} Lage von C bzgl. Koord. A: $H_{Ges} = H_A * H_B$

Freiheitsgrade: $F = \text{Bewegungsfreiheitsgrad} = \text{Summe der Gelenkfreiheitsgrade} = f = \text{Freiheitsgrad des Endeffektors} = 6$ im 3D

Geometrisches Modell:

- Grafische Adrst., Kollisionserkennung, Kräfte ausrechnen (umschliessender Quader, Linienmodell, Volumenmodell)

Kinematisches Modell, kinematische Kette, geschlossen, offen

Zweck: Zusammengang zw. Gelenkwerten und Stellungen, Erreichbarkeitsanalyse

- Denavit-Hartenberg-Algorithmus:

z-Achse immer = Rotationsachse.

x-Achse liegt auf kürzester Verbindung zwischen beiden Gelenk-Achsen, falls die Gelenkachsen sich schneiden:

x-Achse so das die vorige möglichst wenig gedreht werden muss

DH-Parameter:

- zuerst Drehung um Winkel $Teta_i$ (Achtung: $Teta_i$ von Gelenk-stellung $Delta_i$ abhängig!) um alte z-Achse, so das alte und neue x-Achse parallel.

- Dann Translation um d_i auf alter z-Achse bis zu Schnittpunkt mit a_i .

- Dann Drehung um Winkel $Alpha_i$ um neue x-Achse, so das alte und neue z-Achse parallel.

- Dann Translation um a_i : kürzeste Verbindung zwischen beiden Gelenkachsen, auf neuer x-Achse.

Also DH-Matrix von Gelenk i-1 zu Gelenk i: $A_{i-1, i} = R_{z_{i-1}}(Teta_i) * D_{z_{i-1}}(d_i) * R_{x_i}(Alpha_i) * D_{x_i}(a_i)$

Gesamte Vorwärtskinematik (direkte Kinematik) des Roboters (Gelenkwinkel bekannt -> wo ist meine Hand?):

Alle DH-Matrizen in der Reihenfolge der Gelenke Multiplizieren:

$S_{ges}(Delta_vektor) = A_0 1(Delta) * A_1 2(Delta) * ... A_n - 1, n(Delta)$

Inverses Kinematisches Problem (Gegeben: Lage als 4x4 Matrix T, Gesucht: Gelenkwinkel als Tupel $Delta_vektor$)

- Algebraische Lösung:

$S_{ges}(Delta_vektor) = T$, Komponenten-vergleich.

relativ Allgemein, Rechenaufwendige Suche nach Lösung

-Geometrische Lösung:

Anwendungsspezifisch, Basiert auf Trigonometr. Beziehungen, Formel fertig -> nur einsetzen, also schnell und genau

-Numerische Lösung:

bekannt: momentane Position des Arms (Startwert), Zielposition, aktuelle Gelenkwinkel.

Gesucht: Gelenkwinkel für Zielposition.

Vorgehensweise: Aufstellen der Jacobi-matrix des 6-dimensionalen Beschreibungsvektors, Veränderung der aktuellen Position in Richtung Zielposition um Δ_{pos} , Mittels alter Jacobi-Matrix und neuen Zielpositionen Bestimmung der Gelenkwinkeländerung, ...

Hohe Laufzeit, Anfällig für Singularitäten, Es wird irgendeine Lösung gefunden (nicht alle)

Allgemeine Probleme:

Mehrdeutigkeit,

zulässigkeit der Mathematischen Lösung (gelenkwinkelbegrenzung, gegenseitige behinderung,)

Existenz (Ziel im Arbeitsraum?)

Singuläre Stellung

Dynamisches Modell: benötigt für: Systemauslegung, Trajektorienoptimierung, Modellbasierte Regelung.

Kraft-Momenten-Vektor: $Q = M(q) \cdot \ddot{q} + n(q, \dot{q}) + g(q) + R \cdot \dot{q}$

Trägheitsmatrix, Zentrifugal/Coriolis-komp., Gravitationskomponenten, Reibungskräfte

Direktes dyn. Problem:

aus Kraft/Momenten-Vektor und Anfangsgeschw./Beschleun./Lage Geschw./Beschl./Lage in der Zukunft bestimmen.

Inverses dyn. Problem:

aus Geschw./Beschl./Lage die wirkenden oder von den Motoren zu überwindenden Kräfte berechnen.

Lagrange: $Q_i = \frac{d}{dt} \frac{dL}{dq'_i} - \frac{dL}{dq_i}$, wobei $L = E_{kin} - E_{pot}$

Kinetische und Potentielle Energie sind mit DH-Matrizen zu bestimmen.

Newton-Euler

Ermittlung der Geschw. etc durch kinematische Berechnung bis zum letztem Armglied.

Dann Freischneiden des letzten Gliedes, Ermittlung der Kräfte und Momente im Schwerpunkt, Ermittlung der Kräfte und Moment im Gelenk, reduzieren des letzten Gliedes auf diese Kräfte, weiter machen mit vorletztem Glied

Lagrange: +: Einfaches aufstellen der Gleichungen, analytisch auswertbar, geschlossenes Modell.

-: Sehr umfangreich (n^4), nur Antriebsmomente

Newton: +: beliebige Gelenkanzahl, belastung der Armelemente werden berechnet, $O(n)$

-: Rekursion

ZUSAMMENFASSUNG: Direkte Kinematik: Gegeben seien alle Gelenkstellungen. Wo ist der TCP?

Inverse Kinematik: Gegeben ist die TCP-Stellung. Mit welchen Gelenkstellungen kann diese Stellung erreicht werden?

Dynamik: Welche Kräfte/Momente müssen die einzelnen Gelenkantriebe aufbringen, um den TCP mit einer bestimmten Beschleunigung zu bewegen.

Bahn-Planung: Wie sieht eine gute Trajektorie aus, welche alle Kollision vermeidet?

Bewegungsplanung: Position vom TCP als Funktion der Zeit.

Control (Regelung, Steuerung): Modell des Manipulators ist nicht exakt. Was kann man machen, damit trotzdem eine gute Performance garantiert werden kann?

Bahnsteuerung:

Bekannt: Startzustand, Zielzustand. Gesucht: Weg dahin.

-im Gelenkwinkelraum(PTP): asynchron, synchron, ohne dass man sich Gedanken über die Bewegung im kartesischen macht.

Vorteile: Gelenkwinkeltrajektorie einfach, keine Probleme mit Singularitäten. $q(t)$ =Gelenkwinkelvektorsequenz

-Rampenprofil (wird bei kurzen Strecken zu Dreieck): $s''(t) = b_m$

-Sinoidenprofil: $s''(t) = b_m \cdot \sin^2(\pi/t_b \cdot t)$

Anpassung der Zeiten t_b, t_v, t_e , so dass sie Vielfaches vom Interpolations-Abstand sind, und dann Neuberechnung von v_m, b_m

-bei Rampenprofil: Auf/Abrunden

-bei Dreieckprofil: Abrunden, damit man nicht zu spät mit Bremsen anfängt

-bei synchronen PTP: Aufrunden, damit man nicht eine Zeit ermittelt, in der es das langsamste Gelenk garnicht

schaffen kann.

Vollsynchrone: Beschleunigungszeiten sind auch gleich: evtl. kürzerer Weg im kartesischen Raum, aber: alle Beschleunigungen sind fest vorgegeben.

-im Kartesischen Raum(CP): Interpolation im Kartes., dann inverse Kinematik.

gegeben: 6-dimensi. Beschreibungsvektor (x,y,z,alpha,beta,gamma) für Start und Ziel.

zwischenergebnis: zwischenbeschreibungsvektoren durch interpolation

ausgabe: Gelenkwinkelvektorsequenz durch inverse kinematik.

Linearinterpolation: $p(t) * (p_{ziel} - p_{start}) * delta_s * t$; $delta_s$: (weg/zeiteinheit)

zirkularinterpolation: zusätzlich Mittelpunkt vorgegeben. Benötigt, um saubere kreisbahnen

sägen/zeichnen/schweißen etc zu können.

PTP und CP mit überschleifen: statt die zwischenpunkte anzufahren, wird kurz bevor man sie erreicht auf geschwindigkeit/weg zum nächsten zwischenpunkt umgestellt.

Polynominterpolation/(kubische)Splineinterpolation/Bezier(Bernstein)Interpolation

Bahnplanung Problemklassen: vollst./unvollst./kein Umweltmodell./bewegl. Hindern./bewegl Ziel.

Konfigurationsraum/Arbeitsraumhindernis/Hindernisraum/Freiraum Freiraumberechnung: $O(n^m)$

Planungsverfahren: exact(csg), approximativ(Umwelt approximativ beschr.durch Polygone)

Häufig: Konstruktion von „Straßenkarten“. Mit: Retraktion (voronoi - nimm immer den Weg der von je 2

Hindernissen gleich weit weg ist). Oder mit: Sichtgraph-verfahren. Sichtgraphverfahren: Evtl Kugel/Rechteck verwenden um Hindernisse erst zu vergrößern. exact: bei 2 translatorischen Freiheitsgrade. Kürzester Weg: nicht im R^3 Im „Netz“ (Straßenkarten) Suche z.B. mit A* Algorithmus/Potentialfeld/euklid. Abstand(greedy)

Zellzerlegungsmethoden: (Nachbarschaft dann als Graph darstellen und wie oben Suchen. exakte Zerlegung (nur vertikale Linien). approximativ: Unterteilung in Quadrate/Würfel, falls Quadrate teilweise belegt weiter unterteilung, bis Beispielsweise gröÙe=halbe Robotergröße. Aufbau von Quadree/Octree/hexadectree... Bei dynamischen Hindernissen die später hinzukommen: on-the-fly-unterteilung von großen Zellen

Potentialfelder: Skalarfeld U_{AN} , z.b. 1/quadratischer abstand um Ziel. Skalarfeld U_{AB} : Abstoßung um Hindernisse. Ableiten-> Vektorfeld -> Line suchen. Problem: lokale Minima.

Sensorgestützte Bahnplanung: Benchmarks: Simple, Bottleneck, Star, Trap, Detour, Impossible

Best-First-Algorithmus Schnell/wenig tests Große Open/close liste/aufwand auch bei einfachem Weg

Randomized Roadmap-Algorithmus. gut bei detour/unabhängig von zahl der freiheitsgrade unbekannterechendauert/komplizierte Wege als lösung/star,bottleneck brauchen lange

HD-I-Algorithmus (Breitensuche mit Best-First) Löschen der Close/open-liste, wenn an hinderniss vorbei (also wenn einige konfigurationen in richtung ziel frei. Problem: falls man später doch nochmal umdrehen muss.

schnell/auch im leerem Raum großer Speicherbedarf bei mehr Freiheitsgraden/viele Kollisionstests

geschlossener MSR Wirkungsablauf: Aktuatorik <- Prozess -> Sensorik <----- μP <---

Nichtelektr. Messgrößen umformer -> Wandler -> Sensorelektronik Getriebe <- Wandler <- Leistungselektronik

SENSOR: Einheit die auf Stimulus reagiert. Physikalischer Sensor: die mit elektr. Ausgangssignal reagiert

Ausgangssignal: Strom/Spannung/Ladung, Amplitude/Frequenz/Phase

Klassifizierung: extrinsisch/intrinsisch aktiver/passiver Wandler (passiv: erzeugt Signal, aktiv: wandelt signal)

aktiver/passiver Sensor (aktiv: Umwelt „stimulieren“ (signal aussenden)) Wertebereiche:

binär/skalar/vektor/struktur,muster

elementare/integrierte(mit sensorelektronik)/intelligente(mit μP)

MESSFEHLER: statistisch (unvermeidbar/unregelmäßig) /systematisch(vermeidbar durch sorgfältiges untersuchung)

Absoluter Fehler: richtiger Wert Minus Messwert. Relativer Fehler: abs.fehler/messwert

Mittlerer Fehler bei mehreren (N) Messungen: Varianz/Wurzel(N)

Vertrauensbereich: +-Varianz: 68%, +-2*Varianz:99%

Fehlerfortpflanzung: Schätzung durch Summe der Part.Ableitungen * Fehler (Part. Abl: Gewichtungsfaktor) Faustregel:

Addition: Addition der abs. Fehler, Multiplik.: Addition der rel. Fehler. Nicht große Größen voneinander anziehen, sondern besser gleich nur Differenz messen!

Lineare Regression/korrelationskoeffizient sollte möglichst nah bei 1 sein.

Übertragungsfunktion: Beziehung zwischen ein und Ausgangssignal: ideale, reale Meist linear, eindimensional, aber auch polynom/logarith/exponent. mögl.

Sensitivität: Steigung

Messbereich, Ausgabebereich, Genauigkeit Reale Sensoren immer ungenau, sollte aber innerhalb der Genauigkeit liegen

Kalibrierung: Beispiel linear: nur 2 Messpunkte zur Kalibrierung ergibt evtl. falsch Steigung

Hysterese: unterschiedl. Werte je nach Annäherungsrichtung

Sättigung: Arbeitsbereichgrenze

Wiederholgenauigkeit. (statistischer Fehler ??)

Totband: Bereich, in dem Steigung der Übertragungsfkt (fast) 0 ist.

Dynamische Eigenschaften: Sensorausgabe auch abhängig davon wie lange ein Signal anliegt: Sprungfunktion, Sprungantwort, Impulsfunktion, Impulsantwort, Sinusfkt

Sensoren Beispiele

INTERN: Positionssensor: -Photoelektrisch: Graycode / inkrementell mit Zähler

Beschleunigungssensor: -Kapazitativ: Masse ist Platte in der Mitte für 2 Kondensatoren, an denen Wechselspannung anliegt, $U_m = U_s * \Delta d / d_0$ - in Industrie: Masse verbunden mit ganz vielen solchen Platten in der Mitte zwischen zwei anderen Platten, in mehrere Richtungen.

Drehratenmessgeber (inertial) -Kreisel, one degree of freedom -Sagnac Effekt: Licht wird aufgespaltet und läuft in zwei Richtungen, bei Drehung muss es in einer Richtung größere Strecke zurücklegen -> Phasenverschiebung bis zur Auslöschung. Nur bei hohen Geschwindigkeiten. -Ringlaserkreisel: extrem hohe Genauigkeit -Vibrationskreisel: vibration wird angeregt und durch Drehung verändert-> Widerstandsänderung. billig aber ungenau.

Inertiale Navigation: Koordinatensystem: Fixstern- / Erd- / Fahrzeugfest in Fahrtrichtung / Fahrzeugfest in fester Richtung 2 Methoden: Kreisel in Kardanischer aufhängung, darin Beschleunigungs-sensoren oder: Strapdown: feste Beschleunigungs/Drehratensensoren und Uminterpretation der Sensordaten bezüglich des Weltkoordinatensystems mit Umrechnung anhand der Sensordaten. Zusätzlich: Kombination mit Odometrie

EXTERN: Taktile Sensoren: -Schlauch der zusammengedrückt wird auf Kontakt der durchführt (binär). -Kapazitativ (Plattenabstand) oder Magnetisch mit Hall-sensor -optisch: Skalar(eindrücken) / Vektoriell (Änderung des Reflexionswinkels -> Stelle wo Druck ist) -Drucksensitive Materialien: Leiterbahnen entstehen durch Druck. Dann Gittermuster mit Leiterbahnen drunter und drüber. -> künstliche Haut, exakte Positionsbestimmung

-Dehnungsmessstreifen: Längenvergrößerung/Durchmesser verengung: Widerstandsänderung eines Drahtes, hervorgerufen durch eine Dehnung des Drahtes, proportional zur Dehnung ist. Der Proportionalitätsfaktor (genannt: k-Faktor) ungefähr 2 -Messschaltungen: Viertel/Halb/Vollbrücke. -Biegebalken -Kraft-Momenten-Sensor mit 8 Dehnungsmessstreifen

Näherungssensoren (binär): - Kapazitativ: Metallische über Abstand, Nichtmetallisch zwischen zwei Platten/Änderung der Dielektrizitätskonstante - Spule: Metallische: Wirbelstromverluste entziehen Schwingkreis Energie - Lichtschranken/Lichttaster mit ohne Reflektion

Abstandssensoren (skalar): Größere Entfernungen, und Entfernungsangabe -Laufzeitmessung (Ultraschall oder monochromatisches Licht -interferometrie (Überlagerung gesendete/empfangene Welle) -Aktive Triangulation

PROBLEME: Mehrfachreflektion, Wegreflektion, Absorption

-Passive Triangulation (CCD-Kameras, wie Auge von Mensch)

CCD-Kamera: wie Photoapparate Belichtung-> Aufladung von Kondensatoren-> Messung der Ladung CMOS-Kam: Photodioden liefern kontinuierliches Signal, Logarithmische Empfindlichkeit -> Besser bei großen Lichtunterschieden (keine so starke Blendung)

GPS: Satelliten senden ihre Uhrzeit und ihre Position, aus 4 Satelliten Ermittlung der einzigen (2) möglichen Positionen.

AKTUATORIK Energiesteller/Leistungselektronik - Energiewandler - Getriebe

Primäre Energie != Steuergröße (Leistungselektronik) -elektrische, strömungs, thermische, chemische

Elektromotoren, bürstenbehaftet/bürstenlos/Gleich/wechselstrom/schrittmotoren klein, gut regelbar, wiederholgenauigkeit <-> wenig Kraft

Getriebe:Stirnrad/Planeten/Schnecken/Spindel u. Mutter/ Harmonic Drive Stirnrad: Untersetzung: $N1/N2$,
Drehmoment $2=Drehm.1/Untersetz.$

Wirkungsgrad von Motor und Getriebe Multiplizieren sich -> oft unter 50

Motorsteuerung: Mikrokontroller zu schwach für Motorströme, daher Relais/große Transistoren (4 Stück, um auch Richtung zu steuern). Relais nur binär /Transistoren auch Drehzahl. Drehzahlregelung oft mit PWM (Pulsweitenmodulation)

Dehstoffeilemente: Druckbehälter mit Dehstoff der bei Erwärmung ausgedehnt wird und so Kolbe rausdrückt, kombination mit Rückholfeder.

Elektro/Magneto-rheologische Flüssigkeiten: Viskosität von angelegter Spannung/Magnetfeld beeinflusst. Anwendung: Kupplung/Ventile für Hydrauliksystem/Stoßdämpfer

Piezelektrischer effekt: Verformung des Kristall erzeugt Spannung, inverser: Spannung erzeugt Verformung schnelle Reaktion, für kleine exacte Auslenkungen, Bauform: Stapel, Hebelgetriebe Raupenprinzip Hybridwandler: Gropos. mit Magnetantrieb, Festklemmung durch Piezo-El.,Feinpositionierung durch Piezoelement. Weitere Anwendung: z.B. LWL-Anschluss. Ultraschall erzeugung. Magnetostriktiv: Ähnlicher Effekt, jedoch braucht Magnetfeld immer Energie, während Spannungs-feld erhalten bleibt wenn sich nix bewegt. Vorteil: keine Bewegte Elektrode/Temperatur unempfindlicher

Formgedächtnislegierungen: einmaliger Memory-Effekt <-> hin und herschalten durch erwärmung/abkühlung zw. 2 Zuständen. Arbeit wird nur beim Aufwärmen verrichtet. Unterschied zu Bi-Metall!

ANALOG-DIGITAL-WANDLUNG

Operationsverstärker: verstärkt Differenz $U_+ - U_-$ zwischen zwei Eingangsspannungen sehr stark, hoher Innenwiderstand. Bei Rückkopplung U_a an U_- : regelt sich ein: $U_- = U_+$ und $I_- = I_+ = 0$

-Schwellwertschalter

-Summierer: (U_+ an Masse, U_- Rückgekoppelt) $U_a = R_{rck}/R_1 * U_1 + R_{rck}/R_2 * U_2$

-Subtrahierer: (Spannungsteiler an U_+ , U_- Rückgekoppelt)

-Integrierer: Rückkopplung über Kondensator -> durch Op wird Kondensator immer mit gleichem Strom aufgeladen.

AD-Wandler: -Wägeverfahren vergleich nacheinander mit $1/2$ Referenzspannung für 1. Bit, $1(3)/4$ für 2. Bit usw., Zeit Unabhängig von Spannung, aber pro Bit ein Vergleich oder Takt nötig -> Langsam.

-Integrierende Wandler: kontinuierliche Integration (Charge-Balancing) (konstante/keine konstante Ausgangsspannung), diskontinuierliche Int: Dual/Multi-Slope

-DUAL-SLOPE: feste Zeit Aufwärtsintegration mit U_x , dann Abwärtsintegration mit U_{ref} bis 0V. Genau,unkritische Schaltungskomponenten,langsam,schlechte Auflösung

-U/F-Wandler: Wandelt in Frequenz um, Single-Slope bis feste Spannung

-Charge-Balancing: kontinuierliche Aufwärtsintegration mit $U_x + / - U_{ref}$ Zählen wie oft Ladung des Kondensators zu klein ($U_a < 0$).Wandlerdauer unabhängig von U_x , aber lange, nur 2 Referenzspannungen, aufwändige Digioale Elektronik.

-PARALLELE Wandler: Alle Vergleiche die beim Wägeverfahren denkbar wären gleichzeitig (2^n). Mit Und/Nicht-Gatter herausfiltern des größten Vergleichs der 1 liefert (also U_{ref} größer).Dann Wandlung in n-Bit Dualzahl. Sehr schnell. nur 2 Takte, dafür sehr viel Hardware nötig.

-DELTA-Modulation: Wandler kodiert nur Spannungsänderung. z.b. Differenz zwischen 2 Abtastzeitpunkten. Datenreduktion.z.B. 1-Bit-kodierung ob größer oder kleiner geworden. evtl. Tiefpass vorschalten.

DIGITAL-ANALOG-WANDLER: -Primitiv: Addierer mit einer Referenzspannung, die mit n Widerständen: $R, 2R, \dots, 2^n R$ n mal anteilig aufsummiert wird. Problem: Genauigkeit der Widerstände von 2^n für $n > 4$ nicht möglich. Darum Reihenschaltungen.(12-Bit-Wandler der R-2R Kettenleiter)

REGELUNG Lehre von der selbsttätigen gezielten Beeinflussung dynamischer Prozesse während des Prozessablaufs

Grundsituation: Forderung nach selbsttätiger, gezielter Beeinflussung bei unvollständiger Systemkenntnis, insbesondere bei Einwirkung von Störungen.

Aufbau: Stellgröße y -> Strecke|Störgröße z |dynamisches System -> Ausgangsgröße x
Störgröße nur unvollständig bekannt.

STEUERUNG: -> Steuerglied -> Steuersignal -> Stellglied -> Stellgrößen -> Strecke ->

REGELUNG: Rückkopplung, Führungsgröße/Sollwert, Störungen ändern Regeldifferenz (Ist-Soll) -> Reaktion. Leitrechner/Mensch ändert nur Sollwert.

Blockschaltbild: -> Führungsgröße - Rückführgröße Vergleich: Regeldifferenz -> Regler -> Stellglied -> Strecke -> (Regelgröße) -> (Rückkopplung zur Rückführgröße über Sensor)

Beschreibung/Analyse Regelungstechnischer Systeme: Differentialgleichung -> Linearisierung mit abgebrochener Taylorreihe Trick: Integraltransformation in Bildbereich (algebraisch) -> Lösen -> Rücktransformation (Transformation meist mit Tabellen)

Integraltransformation: $F(s) = \int f(t) * K(s, t) dt$, Laplace: $K(s, t) = e^{-st}$, Grenzen: 0..inf. Linearität, Faltung -> Multiplikation, n-te Ableitung -> $s^n *$

Blockschaltbilder: Serienschaltung -> Multiplikation, Parallelschaltung -> Addition, Rückkopplung: $G = G/(1 + -G_1 * G_2)$

Störverhalten: Führungsgröße 0, nur Störungen / Führungsverhalten: Störgröße 0

Steuerung kann als Blockdiagramm modelliert werden, Laplace-Transformation -> Stabilitätsbetrachtung anhand Übertragungsfunktion im Bildbereich.

Verhaltenssimulation durch Übertragungsfunktion im Zeitbereich ausrechnen.

Übertragungssysteme:

proportional, P-Verhalten: (fehlerbehaftet, konstant)

integrierend, I-Verhalten: (schwing um gewünschten Wert)

differenziell, D-Verhalten (dirac-impuls bei Änderungen)

Es gibt P, PI, PD, PID-Regler.

Reglerklassen:

-Kaskadenregler (Sensor zur Strom/Geschwindigkeit/Positionsregelung zurückgekoppelt)

-Kennlinienregler (z.B. zweipunktregler mit einfacher Kennlinie: bis *temp_{wunsch}* an, danach aus (Herdplatte))

-Zustandsregler („gegenteil zu kaskade“: Sensor für mehrere Zustandsgrößen, z.B. Auslenkung, Geschwindigkeit, Beschleunigung ... -> zurückkoppeln in ein Regler der daraus das Beste macht).

-Adaptive Regelung: Zusätzliche Rückkopplung über Gütekriterium und Adaptionsvorschrift.

BEISPIEL: (Dezentrale Gelenkregelung von Roboter, interne Regelung: Kaskadiert Geschw.+Lage, externe Regelung: kartesisch, Jacobi-Matrix (dynamisches Modell))

(Bei Kaskadenregler: Übertragungsfunktion für innersten Regelkreis finden und dann mit der Ü-Fkt. und dem nächstäußeren Regler weitermachen (Ersatzregelkreis)).

-Kraft/Positionsregelung: Interaktionskräfte berücksichtigen.

-Impedanzregelung

Steuerung/Regelung durch Echtzeitrechner: Rechner nur parallel zur normalen Regelung (Ausfallsicherheit), Rechner berechnet neue Führungsgröße aus Istwert

oder Rechner ist Regler mehrerer Größen. Flaschenhals. z.B. Kameraüberwachung. ein Programm für viele Regler.

UNTSCHARFE REGELUNG Klassische R.: Benötigt Modell der Strecke, analytisch.

Fuzzy-R.: Regelmenge, intuitiver Entwurf, kein vollständiges Modell der Strecke nötig, Robust, intuitiv, erweiterbar

Fuzzymengen: Charakteristische Fkt. kann alle Werte zwischen 0 und 1 annehmen, μ ordnet jedem Element x der Grundmenge einen Zugehörigkeitsgrad zu. GANZ ANDERS als Wahrscheinlichkeiten.

Träger, Kern, Niveaumenge, Teilmenge

Unschärfe Logik: Konjunktion (t-Norm): Axiome: Kommutativ ($t(a, b) = t(b, a)$), Assoziativ, Monoton, Beispiele: Minimum, Produkt.

Dijunktion (s-Norm): Komplement von t-Norm mit deMorgan ($s(a, b) = 1 - t(1 - a, 1 - b)$)

Komplement 1-a, Durchschnitt (t-Norm), Vereinigung (s-Norm),

Kartesisches Produkt: bei unterschiedlichen Grundbereichen: $\mu_{A_i \times \dots}(x_i \dots) = t(\mu(x_i) \dots)$

Inferenz: Implikation als Fuzzy-Relation gegeben, Konklusion: $\mu_{res}(y) = s_{x \in X} \{t(\mu_{praemisse}(x), \mu_{implikation}(x, y))\}$

FUZZY-REGELUNG: Fuzzifizierung der Eingangsgrößen

Schauen welche Regeln Aktiv sind

Grad des Zutreffens einer Regel

Finden der Schlussfolgerung jeder Regel

Zusammenfassen der Ergebnisse aller aktiven Regeln

Defuzzifizierung

Fuzzifizierung: Linguistische Variablen (z.B. Temperatur) und Linguistische Terme (z.B. sehr kalt, kalt, warm, heiß,

sehr heiß) oft Standardaufteilung in 5 Terme mit jeweiliger Zugehörigkeitsfunktion .
Sympathievektor bilden.

Regeln. Wenn linguistischer Term und linguistischer Term und DANN linguistischer Term

Grad des Zutreffens einer Regel: Verundung des Zutreffens aller Prämissen der Regel.

Schlussfolgerung einer Regel: Konklusion bilden: Abschneiden der aktiven

Regelschlussfolgerungszugehörigkeitsfunktionen auf Höhe der Prämissen und Vereinigung dieser Funktionen.

Defuzzifizierung: Auswahl des „richtigen“ Wertes, so dass obige gebilde Funktion (z.b.) Maximal. falls mehrere

Maxima: Mittelwert von allen/linkester/rechtester. Oder Schwerpunkts-Methode.

DIGITALE SIGNALVERARBEITUNG

Eingangssignal->Abtastung->digitale Verarbeitung->Rückführung->Ausgangssignal

Abtastung:Tiefpass,S&H-Verstärker,A/D-Wandler ; Rückführung: D/A,S&H,Filter(glätten)

Signaldarstellung im Frequenzbereich: $s(t)$ =Summe von Sinus und Cosinus-Fkt, oder Summe von
Phasenverschobenen Cosinus-Fkt. mit $\cos(x) = 1/2(e^{jx} + e^{-jx})$ und mit Amplituden $c_{-n} = c_n$ und
Phasenverschiebungen $\varphi_{-n} = -\varphi_n$ und komplexer Fourierreihe $d_n = 1/2c_n e^{j\varphi_n}$ ergibt sich $s(t) = \sum_{-\infty}^{\infty} d_n e^{jn\omega_0 t}$

Somit ist jedes periodische Signal durch ein diskretes Amplitudenspektrum, Phasenspektrum und die
„Grund“-Frequenz ω_0 beschreibbar.

Bei aperiodischen: ω_0 unendlich klein -> Übergang von Summe zu Integral -> Fourier-Transformation:

$$S(\omega) = 1/(2\pi) \int s(t) * e^{-j\omega t} dt$$

Abtastprozess: maximale Abtastrate gegeben durch Wandlergeschwindigkeit, minimale durch Abtasttheorem:

$$f_a > 2f_g$$

Abtasttheorem: Wenn nicht erfüllt überlappt im Frequenzbereich das Signal mit sich selbst (nach Faltung mit
Abtast- „signal“)

SIGNALVERARBEITUNGSPROZESS (DSPs, Filter).

Rückführungsprozess: Glitches: bei Wechsel zwischen zwei digitalen Werten können Glitches auftreten, darum
S&H-Verstärker, welcher Ausgang für eine Takt-dauer festhält.

Filter: eckiges Signal (also hohe Frequenzen) -> Tiefpass zum Glätten.

Rauschquellen: Rundungsrauschen (nur 32 Bit-> runden), Quantisierungsrauschen (endliche Auflösung,
Abtastfrequenz), Übersteuerungsrauschen (Überschreitung des

Wandlerbereichs->Verzerrungen), Untersteuerungsrauschen (Leerkanalgeräusch, Ausklingendes Signal)

DIGITALE FILTER: Früher: hoch,Tief,Bandpass durch RLC-Glieder, Operationsverstärker

Digitale Filter:(Aufwändiger aber unabhängig von Bauteileigenschaften,externen Einflüssen, Weniger
Störungen,leichter zu ändern, programmierbar, adaptive Filter leicht (Modem: Filter an Leitungsqualität
automatisch anpassen))

Anforderung an Rechenleistung: 1Mhz Abtastrate, 10 Befehle/zyklus nötig -> 10MIPS Prozessor nötig. Für extreme
hohe Frequenzen deswegen noch Analogfilter. Digitales Fernseh: $f_a = 13,3\text{Mhz}$.

FIR-Filter: Bilde Ausgangssignal durch Linearkombination aus aktuellem Wert, und verzögerten Werten.

IIR-Filter: Rückkopplung, Ausgangssignal Linearkombination aus aktuellem Wert und rückgekoppelten (verzögerten)
Werten.

RECHNERARCHITEKTUR

ASIC (application spezifik integrated circuit) (allgemein verfügbar aber anwendungsabhängig, generierbar, bibliothek)

DSP (digitaler signal prozessor) (anwendungsspezifisch, fest konfiguriert, komplett verfügbar)

PLD (programmable logic device) (anwendungsunabhängig, feldprogrammierbar,)

SPS (speicher programmierbare steuerung)

Microcontroller (anwendungsunabhängig, core fest)

Industrie-Pc

DSP: spezialisiert auf häufige Operationen der digitalen Signalverarbeitung(Matrizenmultiplikation),
anwendungsspezifische Adressierung (für FIR/IIR-Filter, je Zyklus wird ein Abtastwert weitgeschoben), eingeschr.
Parallelität : Zuweisung in mehrere Register in einem Takt., Saturation arithmetik, Heterogene Registersätze,
Echtzeitfähig.

Zum Beispiel: 64K Programm+64k Datenspeicher, 64k E/A-speicher, 32-bitALU schreibt in Akku, PLU,

Multiplizierer, Timer, Interrupts

PLD: kürzere Entwicklungszeiten und billiger in kleiner Stückzahl als ASIC, Vielseitiger, einfacher und schneller Herzustellen. größer als IC. (1/100 Kapazität, 1/3 Geschwindigkeit)

es gibt: maskenprogrammierbar (MP), in Anwendung programmierbar (z.B. durch EPROM/SRAM) (FP)

Architektur: (MP/FP)-Logikzellen, verbindungen dazwischen durch eprom/sram schaltbar, umprogrammierung im Betrieb möglich. ungleiche Signallaufzeiten.

CPLD: gleiche Signallaufzeiten, jedes signal muss horizontale und vertikale Leitung einmal komplett durchqueren, keine schalter dazwischen.

SPS/VPD(verbindungsprogrammierte-): zyklische Abarbeitung logischer Verknüpfungen/Steueranweisungen, OS kopiert Eingänge in speicher, Programm wird abgearbeitet, Ausgänge werden geschaltet. Zykluszeit bekannt, maximale Reaktionszeit = 2x Zykluszeit.

verknüpfungssteuerung (boolche)/ablaufsteuerung(schrittweise Ablauf, Bedingungen)

Kontaktplan/Funktionsplan/Anweisungsliste

MicroController: Micorcomputer+Speicher+Peripherie in einem Baustein

Geringer Platzbedarf, billig, wenig zusätzliche Bauteile nötig, enthält serielle/parallele Schnittstelle, A/D und D/A-Wandler, Programm/Daten-Speicher

Bsp: Intel 8051. ca 128kB speicher, timer, interruppts, spezialregister

Industrie-PC: robust (temperatur/schock/emv), x86-basiert, mit Betriebssystem(windows/linux/qnx)

Bauform: tragbar, slot, hutschienen, PC-104: 40+64 Pinx, 10x10 cm

Kommunikation

Busse: Mbyte/min ... Backbone(LAN/WAN)/Zellenbus(heute Ethernet) / Feldbus / Sensor-Aktor-Bus ... Bit/ms

Anforderungen: robust, fehlertoleranz/diagnose, wartbarkeit, Echtzeit, Latenzzeit, Ereignisbasiert(interrupts)

OSI: Bitübertragungs (Taktrückgewinnung etc)

Sicherungs (Fehlererkennung/korrektur)

Vermittlungs (routing/adressierung)

Transport (logische Kanäle)

Sitzungs (auf und Abbau von logische Kanäle)

Darstellungen

Anwendungen (in der Regel nur 1,2,7)

BITÜBERTRAGUNGS- Abtasttheorem: Signal der Bandbreite B kann man wiederherstellen durch Abtastwerte der Frequenz 2B, Nyquist: Signal der Bandbreite B kann nur Abtastwerte der Frequ. 2B kodieren.

Bei V diskreten Sufen: $Datenrate = 2B \log_2 V [bit/s]$

Rauschabstand: $signalleistung/rauschleistung. Datenrate = B \log_2 (1+rauschabstand) [bit/s]$

Probleme: Dämpfung – repeater

Reflexion – abschlusswiderstand

bitorientierte fehlerkorrektur: falungscodes

elektromagnet. starhlung: gegenseitiges aufheben durch verdrehte Leitungen

störimpulse: differenzielle Übertragung: Statt Masse negatives Signal, da störung auf beides wirkt, wird störabstand erhöht

Taktrückgewinnung: FlankenWechsel ausreichend oft -> srampler/bit stopfen(einfügen einer flanke alle x einisen, die gleich bleiben, empfänge entfernt diese flanke wieder)

Codes: Selbsttaktent, gleichstromfrei(übertragung über stromleitung, z.b. telefon)

NonReturnToZero (kein Takt), RetornToZero (flanke für 1, doppelte bandbreite, bei langen 0folgen kein Takt),

Bipolar(abwechselnd plus minus, gleichstromfrei, kein takt), Manchester (flanke für 1 und 0, doppelte bandbreite, takt), 4B->5B codierung (takt).

SICHERUNGS- starke fehlerbehaftung, meist bündel (error bursts), ethernetverdratung: pro sekunde ein fehler

Redundanz nötig. ERKENNEN: Echokontrolle, paritätsprüfung, CRC

KORRIGIEREN (in stark gestörter Umgebung): Hammingcode (mit Abstand 3 1bit-korrigierend XOR 2-erkennend), Faltungscodes

CRC: polynomdivision durch generatorpolynom, rest wird angehängt und mitgeschickt, empfänger dividiert auch -> darf kein rest rauskommen. erkennt bis zu alle fehlerburst bis 16 bit länge.

Stochastik: Zuordnung von Prioritäten, Wahrscheinlichkeiten, Verteilungsfunktion für Schaltzeit(abhängig von Markierung) zu jeder Transition.

Hierarchische P.: 1 Transition kann durch Petriernetz verfeinert werden. Es ergeben sich konstruktoren, die Markierung eines feineren Netzes die Markierung der abstrakten Stelle zuordnen

Partitionierung: geeignete Aufteilung aller SW-Primitive zu Tasks, Minimierung von

Antwortzeiten,ausführungszeit,ressourcenbedarf,kommunikation Allokation: Zuordnung der Tasks zu Recheneinheiten
Scheduling: Ausführungsreihenfolge der Tasks

Allokation: vollständige Ausnutzung der Ressourcen, stark verknüpft mit Scheduling, Minimierung der kommunikation zw. Prozessoren, minmierung der Prozessoreauslastung

graphbasiert:Flussgraph, Fluss: Knoten=0 Quellen>=0 Senken<=0, Trennmengen, min-cut=maxflow

Prozessoren=Quellen/Senken, Tasks: Innere Knoten, E_T verbindungsgraph der Tasks=Kommunikationskosten, E_P Kanten von Prozessoren zu allen Tasks =kosten um Task nicht auf Prozessor auszuführen (Summe von kosten auf allen Prozessoren/($n - 1$) - kosten auf Prozessor wo potentiell nicht ausgeführt wird.

Allokation: Allokationscutset, jeder teilgraph enthält ein prozessor, minimieren.

Heuristisch: Clustering (irgendwie auf prozessoren verteilen), Iterative Verbesserung: Zuordnungen nach Heuristik vertauschen

SCHEDULING (einprozessor/SMP, echtzeitplanung) unterbrechbare, nicht unterbrechbare, periodische, sporadische, statisches, dynamisches, explizites, implizites Scheduling.

jeder Task/Prozess hat: Bereitzeit,Startzeit,Ausführungszeit,Deadline,Abschlusszeit,(Periodenzeit)

Planbarkeitstest,schedule construction,dispatching

brauchbar (falls alle Zeiten/Fristen ohne überlappung eingehalten), vollständiges Verfahren

Planen durch suchen ($O(n!)$,vollständig,nicht unterbrechbar)

Planen nach Fristen (vollständig bei gleichen Bereitzeiten oder bei unterbrechbaren) Planen nach Spielräumen (SMP, NP-vollst. ab 2 Prozessoren, vollständig bei unterbrechbaren, statischen) Rate-monotonic (für periodische Proz.)

Planen durch Suchen: Baum aufbauen, zunächst ohne Fristen zu berücksichtigen, von jedem knoten aus alle Prozesse jeweils als nächsten einplanen. Dann Lösungen streichen, die Fristen einhalten. Bei beachtung von bereitzeiten: jeweils erst zur Bereitzeit des proz. beginnen. findet nicht alle lösungen, da immer so früh wie möglich begonnen wird.

EDF: wird immer prozess zuordnen der als nächstes fälligst. statisch, nicht unterbrechbar (planen, wenn prozess fertig wird)

statisch, unterbrechbar (planen, wenn prozess fertig wird oder neuer bereit)(vollständig)

dynamisch, unterbrechbar (planen, wenn neuer prozess hinzugefügt wird, einsorieren in liste und ersten aus liste nehmen, (und wie bei statisch, unterbrechbar))

periodische prozesse: findet brauchbaren, falls: $\sum_{i \in Proz} \delta e_i / \delta p_i \leq 1$

Planen nach Spielräumen: nicht-unterbrechbare/unterbrechbare, Mehrprozessor: Prozesse können jederzeit Prozessor wechseln, Tupel(Prozessnr,startzeit,rechenzeit,prozessornr.)

Planen nach Fristen berücksichtigt nicht, bis wann prozess spätestens anfangen muss, um noch fertig zu werden -> problem bei mehrprozessor.

SPIELRAUM = Deadline - bereitzeit - rechendauer: DYNAMISCH: verteile immer prozesse mit kleinsten m spielräumen auf die m Prozessoren.

Grafische Darstellung durch Spielraumachse, Rechendauer-achse: in jedem Schritt bewege m prozesse mit kleinsten spielräumen entlang der rechendauer achse (bis rechendauer = 0, danach prozess weg), und alle andren entlang der spielraumachse.

STATISCH: für jede Deadline eine prozess-klasse PD_i , tabelle mit: req_i (Prozessklasse) gibt an wieviel recheneinheiten von prozess i bis zu dieser deadline wenigstens berechnet werden müssen. Auch als Graph mit Quelle -rechendauer-> prozesse -deadlineverlängerung-> prozessklasse -m*deadline-> senke darstellbar.

FÜR DYNAMISCHEN FALL: NICHT VOLLSTÄNDIG

Planen nach monotonen Raten: rms(i): prioritäten des prozess i, propotional zur frequenz-1/periode (also monoton). Es darf immer der Prozess mit höchster priorität sofort rechnen, wenn er bereit ist, dann der mit zweithöchster usw. Sehr einfach zu implementieren, aber nicht vollständig.

kritischer Zeitpunkt: wenn Aufgrund der bereitzeiten der andren prozesse ein prozess so spät wie irgend denkbar fertig wird. kritisches intervall in dem Fall die rechendauer dieses prozess.

RMS ist brauchbar, falls kritisches intervall immer in jeweilige periode passt.

RMS wenigstens so gut, wie alle andren Verfahren die auf priorität basieren

RMS vollständig, falls prozessorauslastung $U(n) < n \cdot (n\text{-te Wurzel}(2) - 1)$

Scheduling NP-hart bei nicht-unterbrechbaren und Mehrprozessorsystemen (bsp knappsack bei planen nach spielräumen).

ROBOTERARCHITEKTUREN Fähigkeiten: Erfassen-lernen-planen-ausführen-motorik-ausnahmebehandlung

Klassifikation: funktionsorientiert/verhaltensorientiert; hierarchisch/verteilt

hier. fkt.orient.: Hierarchieebenen für die versch. Funktionen

Oft: Sensorik(G) <-> Weltmodell(M) <-> Planung(H) als Funktionen, die jeweils innerhalb einer hierarchieEbene miteinander kommunizieren.

Auteilung in 4 bis 6 Ebenen., H-Modul: Komm. von oben nach unten, Verfeinerung der Befehle mit Hilfe von Informationen des Weltmodell, informiert Weltmodell über geplante Änderungen;

G-Modul: Kommunikation von unten (Sensoren) nach oben (ausgewertete Sensordaten), Übermittlung der Informationen an das Weltmodell, nutzt Planungsinformationen und Weltmodell um Unterschiede zu finden (zwischen Erwartung und Realität)

M-Modul: Referenzdaten, Vorhersage, Benutzerschnittstelle

Implementierung jedes Knoten: als endlicher Automat, Zustandsübergang löst Aktion aus, Ein/Ausgabepuffer,

H-Modul: Eingabe: Anweisung, Sensordaten(Weltmodell) -> Ausgabe: Teilaufgaben/operationskette

komplexe aufgabe->taskebene->einfache elementaroperationen->geom. trajektorie->gelenktrajektorie

zerlegung in (parallel/sequentiell abgearbeitete) Teilpläne, parallel laufende pläne kommunizieren auf niedrigerer hierarchieebene evtl nochmal.

Beispiel Montagesequenz endl. Automat

G-Modul: Signal -> vergleich mit vorhersage->zeitliche integration der abweichungen,->räumliche integration mit mehreren Sensoren-> Weltmodell aktualisieren.

M-Modul: ein einziges Modul mit Schnittstellen für verschiedene Hierarchieebenen der G und H-Module

verteilt fkt-orientiert: zentraler kommunikationsmechanismus(blackboard), über den alle Fkt verbunden sind

hier. verh.orient.: hierarchie von einfachen zu komplexeren Verhalten (halte hand ruhig -> bring vase von a nach b), kompetenzebenen z.B. in subsumtionsarchitektur, multiple goals, multiple sensors, robustness, additivity

verteilt verh.orient. z.B. roboterfußball (menge von unabhängigen teilsystemen mit identischen verhaltensmustern), multiagentensysteme (selbstorganisation, assoziative speicherung von informationen, erkennung von bestimmten mustern): koordinierung erfolgt über verhaltensmuster.

Beispiel: comros: strategische, taktische und reflexive Ebene, Elementaragenten: autonom, entscheidungseinheit <-> sensor-aktor-zyklus, strategische ebene: gesamtplanung

ENDEFFEKTOREN Scheren/Zangen/Magnet/Saug-Greifer, vergrößerung von Greiffläche/Formschluss, Hände

Greifsequenz: Bewege Hand in Anrückposition, Fahre in Greifposition, Schließe Greiffinger, Fahre in Abrückposition, Bewege Hand in Anrückposition von Teil B, MontagePosition, verbinden, öffnen, Abrückposition.

Bewegungstypen: Greifen/Loslassen: Sicherer Griff, Vermeidung von Kollisionen mit Hand

Anrücken/Abrücken: Planung (Position/Orientierung), Vermeidung von Kollisionen mit Arm

Anrücken/Abrücken mit Objekt: Kollisionen von gegriffenem Objekt

Verbinden. Transferbewegungen (schneller aber ungenauer)

Interne Bedingungen: Gültigkeit(greifmerkmale von objekt und hand überlappen)/Kollisionsfreiheit/Zugänglichkeit

Externe Bedingungen: Kollisionsfreiheit/Kinematik (Arbeitsraum)/Stabilität (Griff, Scene)

Fingerspitzenkontakte: Punktkontakt ohne Reibung(Kraft normal zur Fläche); starrer Punktkontakt mit Reibung(Kraft normal und tangential); nicht starrer...(normal+tangential, rutscht)

Wrenchvektor: Kräfte und Momente in Kontaktpunkt $(f_x, f_y, f_z, \tau_x, \tau_y, \tau_z)$. Aufteilung in w_n, w_t, w_τ mit normalen, tangentialen Kräften und axialen Momenten. Bei m Fingern erhält man so die 6x3m Greifmatrix aus

Wrenchvektoren.

GRIFFE:

Gleichgewichtgriff: Summe aller Kräfte und Momente ist 0. in Greifmatrix G: Alle Normalkräfte größer 0, Reibung druck Druck ist größer der Tangentialkräfte/Momente und für ein externes Moment e gibt es ein c mit: $G \cdot c + e = 0$

Kraftgeschlossene Griffe: für ALLE e gibt es ein c mit $G \cdot c + e = 0$. Es können also Kräfte und Momente in alle

Richtungen aufgefangen werden. ohne Reibung, 2D: 4 KP. 3D: 7-12 KP; mit reibung: 2D: 3 KP, 3D: 4 KP
Formgeschlossene Griffe: ausschließlich abhängig von Position der Kontaktpunkte: nur Kraft normal zur Fläche zählt.
Modifizierte 6 x m Greifmatrix G' : enthält nur w_n normale Wrenchvektoren. Trotzdem muss noch gelten: für ALLE e gibt es ein c mit $G' * c + e = 0$. 2D: 4 KP, 3D: 7 KP.

Stabile Griffe: zusätzlich Modellierung nicht starrer Finger: wird Objekt aus gewünschter Lage bewegt, wird es wieder zurück in Ursprungslage gedrückt. Spezifikatin durch Potentialfunktion in Abhängigkeit von Lage und Orientierung des gegriffenen Objekts. Griff Instabil, falls Lageänderung existiert, so das Potential (potentielle Energie) geringer wird.

Suchraum: Freiheitsgrad des Roboters (6) + Konfigurationsparameter der Hand. Symbolische Griffe:

Greifmerkmale(Flächen/Kanten/Ecken) die mit Greiffingern in Kontakt treten.

Greifbedingungen: parallele/lokale Zugänglichkeit der Kombination mehrere Greifmerkmale, öffnungsweite des Greifers.

Objektmodell: Oberflächenrepräsentation durch Flächen/Kanten/Ecken, Objektbaum

Scenenstabilität: Es darf keine Verschiebung der nicht befestigten (gegriffenen) Objekte geben, die die Potentielle Energie verringert.

ROBOTERPROGRAMMIERUNG:

On-Line: Einstellen, Teach-In, Play-Back (Programmieren durch Beispiele): keine Programmierkenntnisse erforderlich, keine zusätzlichen Rechner, Arbeitsraum muss nicht vermessen werden (nur Gelenkkordinaten werden gespeichert), Störgrößen werden automatisch berücksichtigt. Kein Einbezug von Sensoren, keine Korrektur von Bahnen, keine intelligenten Roboter

Off-Line: Text, CAD, Simulation

Einstellen: Früher: durch mechanische Schalter oder Stopper, Steuerung muss nur noch richtige Stopperstellung „auswählen“, keine sehr freie Programmierung

Teach-In: Steuerungsgerät (Joystick, Maus, Teach-Kugel) um Roboter wie gewünscht zu positionieren.

Playback: Manuelle Führung des Schwerkraftfreien Roboters. (schwierig, gefährlich, kaum benutzt, kein Platz)

Master/Slave System: (teuer, Einsatz bei Teleoperation, Rückmeldung von Kräften)

CAD: Softwaresimulation von Robot und Zele: sehr aufwändige Simulation, Gefahr falscher Simulation, wenns klappt ist sehr gut und einfache Optimierung

Programmieren durch Training: Forschungsgebiet: Vormachen, was der Roboter dann nachmachen soll, Roboter macht solange nach, bis es gut genug ist.

Roboterorientierte Programmierung: Roboterprogramm mit Bewegungsbefehlen: „Fahre auf einer geraden Linie nach Punkt B“. Roboterspezifische Datentypen (Vector, RotMatrix, Transmatrix, Gelenkwinkelpositionsvektor, Frame für Koordinatensysteme). Interrupts müssen möglich sein

kartesische Koordinaten zugelassen, so kann das Programm besser portiert werden und genaue Kenntnis der Kinematik nicht notwendig. Jedoch muss der Arbeitsraum sehr exact vermessen werden.

Aufgabenorientierte Programmierung: höher ebene als Roboterorientierte, nur noch Angabe: WAS muss roboter tun, nicht: wie muss Roboter es tun. intelligente Roboter. Aufgabenplaner erzeugt dann Programm in Roboterorientierter Sprache. Algorithmen zu Greif/Bahnplanung benötigt, Wissensbasis, Umwelt.

PLANUNGSSYSTEME:

Aktionsplanung, Scheduling (Reihenfolgen/Ressourcenplanung), (Bahn/Greifplanung)

Gegeben: Start und Zielzustand.

Gesucht: Menge (sequentiell, nebenläufig, koordiniert-nebenläufig) von Aktionen, um Start in Zielzustand zu überführen

Problem: nicht alle Aktionen immer möglich: Greifen nur falls: Objekt zugänglich, nicht zu schwer, Greifer offen, Greifer am Greifpunkt, Greifer geeignet.

Nebenbedingungen: (optimale) Zeit, wenig Werkzeugwechsel, Genauigkeit...

ARTEN: (nicht/)-linear, monoton (Zwischenpositionen), zentral/verteilt, ein/mehr-stufig (Teilziele)

Grundkonzept: Z Zustandsmenge, O Menge der Aktionen a mit Vorbedingung V_a . $g_a : Z \rightarrow Z$ Zustandsübergang.

Gesucht: Folge M_i mit M_0 Startzust., M_n Zielzust. und für jedes i : M_i erfüllt V_{a_i} und $g_{a_i}(M_i) = M_{i+1}$

LÖSUNG: logische Abbildung

Baum mit Knoten als Zustände und Kanten als Aktionen durchsuchen (Breiten/Tiefensuche, Heuristik)

logische Abbildung: monotones Kalkül finden und in richtige Richtung ableiten.

für komplexe Problemstellung existiert nichtimmer eine streng monotone Reduktionsordnung
implementierung allein als logischer Beweis nicht möglich/sinnvoll

Sensorausgaben = Eigenschaften. $holds(x_j, y)$: Eigenschaft x_j gilt in Zustand y . $result(u, y)$: Ergebniszustand nach Ausführung von Aktion u in Zustand y . $reachable(x, y)$ Zustand x aus Zustand y erreichbar.

so kann man Terme bilden und Beweisen, dass $holds(x_{start}, y_{start}) \Rightarrow \exists y_{ziel}$ mit $holds(x_{ziel}, y_{ziel})$ und $reachable(y_{ziel}, y_{start})$ mit Hilfe von Regeln, die die Aktionen beschreiben.

STRIPS: Prozesszustand als Konjunktion von Formeln die Aussagen über Eigenschaften des Prozess machen.

Operator: 4-tupel aus (name, Bedingung, neueEigenschaften, ungültigeEigenschaften)

Cranfield-Montage-Benchmark. Kriterien: keine Durchdringung, keine Seiteneffekte, Stabilität

AMR:

bewegt sich autonom in Einsatzumgebung um Aufgaben zu erledigen, autonom=fällt selbsttätig Entscheidung (evtl. auch mal falsche aufgrund fehlender information), autark=führt energie mit

NICHT nur: ferngesteuert, stationär, geregelt (autopilot)

Probleme: Hinderniserkennung, umgehen. Lokalisierung. Weltmodell, wie?. Navigation, Bahnplanung,

karten, kollisionsfrei. Objekterkennung: gesuchtes, Landmarken. Objektmanipulation: Greife, Sensorik zum Erkennen der Griffe, Kräfte.

Fortbewegung: Laufen (2 oder x-Beiner), klettern, schlängeln, hüpfen, räder, ketten.

Vort/nachteile: anzahl aktuatoren/sensoren. komplexität des systems/der steuerung. energieeffizienz.

Räder/Ketten: Bewegung senkrecht zur Achse. Lineargeschw. v /Lenkgeschw. ω .

Differentialantrieb: 2 getrennt angetriebene Räder auf Achse in mitte, zusätzliche freie Räder

angetrieben u. gelenktes Rad: 2 feste HinterRäder, ein gelenktes Angetriebenes vorne

Ackermannlenkung: 2 über differential angetriebene Räder auf HinterAchse, vorne 2 gelenkte Räder

oder: 2 gelenkte angetriebene Vorderräder, 2 feste Hinterräder

Synchrondrive: Omnidirektional: viele Räder angetrieben, alle Räder werden gleichzeitig in gleiche Richtung gelenkt.

Mecanumräder: ballige Walzen parallel zur Achse auf Umfang des Rades-> lenken leichter. Nur langsamers fahren.

es fehlt: 2 getrennt lenkbare Achsen, alle angetrieben

Lagebestimmung durch Integration über gefahrene Bahn (Abtasten, diskretisieren, Trapezregel)

Gelenkantriebe: gleichstrommotor mit harmonic drive: +präzise, -gewicht des antrieb

seilzug: unpräzise; doppel/einfachwirkende schubstange: +hohe kraft, -einfache langsam

Getriebe: hydraulikzylinder (kraft/langsam), pneumatikzylinder (federt/nichtlinear), schraubengetriebe (kraft präzise/langsam), linearmotor (präzise, klein/schwach)

Zuggetriebe: Seilzüge, pneumatische muskeln.

SENSOREN: (präzise, geschw., zuverlässig, kosten)

Ziel: erfassung der Umwelt aus partieller, zu verarbeitender Information der Sensoren (Multisensoresystem)

Motorreglung benötigt: Strom, Drehzahl, Winkel

Neigungsmessung (inclinometer): linse in Öl bündelt Licht an verschiedene Stellen je nach Neigung

Orientierungsmessung: Kompass, Kreisel. Murata Kreisel (Piezokristallschwinung, einfach, geringe Genauigkeit), Sagnac Effekt

Entfernungsmessung: Radar/Laser (quadratische Abhängigkeit der empfangenen reflektierten intensität vom abstand), Ultraschall (Laufzeitmessung, nachteil: "weiche" objekte)

Lichtschnittverfahren, stereokamera, strukturiertes Licht, laserscanner mit laufzeitmessung

WEITERE KOMPONENTEN:

Energie: Formen: Elektrisch (24=, 230), Pneumatisch, Hydraulisch, Brennstoffzelle u. Verbrennungsmotor

In Speicher -> rechtzeitig wieder aufladen gehen

Permanent: Sonne, Wind, Kabel, gerichteter Strahl (Laser, Microwelle)

Rechner: hartes Echtzeitbetriebsystem, Interrupts (begrenzt viele!), Peripherieanschluss (USB, seriell...), (Bussystem, adressierung, z.b. kommunikationskontroller liest daten von sender und schreibt sie zu allen empfangern, passgenauer datentransfer, starre synchronisation aber komplexe verwaltung)